By Dennis F. Galletta, Alexandra Durcikova, Andrea Everard, and Brian M. Jones

Does Spell-Checking Software Need a Warning Label?

Users—ironically, often those most verbally armed—put too much trust and little effort in questioning spell- and grammar-checking programs.

Decades ago, as personal computers began to pry their way into our organizational lives, word processing software could barely keep up with fast typists. Today's processors are two to four thousand times their 1MHz speed in 1980, and have data paths eight times their former size. On the road to greater speed, vendors seem to have always rushed in with more sophisticated features to use up those increasingly faster computer cycles. Taking up some of that power are formatting features that provide a close approximation of WYSIWYG editing. Some of that power is also devoted to content features, which are the subject of this study.

Two important content-related features are offered by spelling and grammar checkers (which we will call "language-checking software" in this article for shorthand). In the past, these functions were run in "batch" mode, invoked only after completion of a draft of a document. Today, because language-checking software is run in real time, and is "on" by default, nearly all computer users are accustomed to having their keystrokes monitored as they type. The software combs the text to find misspellings and common usage errors, such as the use of fragments, run-on sentences, subject-verb disagreement, passive voice, double words, and split infinitives. Problems are flagged with colored wavy lines, begging for user attention.

Unfortunately, it does not take long to discover that, while sophisticated beyond the wildest dreams of many users a decade ago, the software is imperfect in important ways. There are false negatives, where the language-checking software fails to detect true errors, and false positives, where the software detects problems that are not errors.

False negatives are troublesome because they might allow users to overlook problems that could be obvious to the human reader. Previous research has shown evidence of false negatives in Microsoft Word 2000. Kies [6] performed an analysis of the 20 most common grammar errors identified (from [2]), and found that Word 2003 uncovered only six of them. Further, its suggestions were incorrect in two of the six cases.

ILLUSTRATION BY JASON SCHNEIDER

Therefore, the language-checking software in even the latest version of the most popular word processor misses the overwhelming majority of the most common false negatives. An example of a false negative used in our study was "Go ahead with the complete role-out," where "role" is not flagged to be replaced by "roll."

False positives are also troublesome, although this issue has not been studied extensively in a usage context. From time to time, perfectly acceptable words or passages are flagged erroneously by language-checking software. In many cases, following the software's suggestions would either distort the true meaning of the sentence or would create an obvious, but unflagged new error. An example illustrates the problem. The software analyzes the sentence "Multiple regression was run," underlines "regression" and suggests it be changed to "regressions." If the user follows that advice, other difficulties cascade down the errant path. The word "was" is then underlined and the suggestion is made to change the word to "were." The final sentence reads "Multiple regressions were run," and, although there are no remaining grammatical difficulties with that version, it distorts the true meaning of the original sentence.

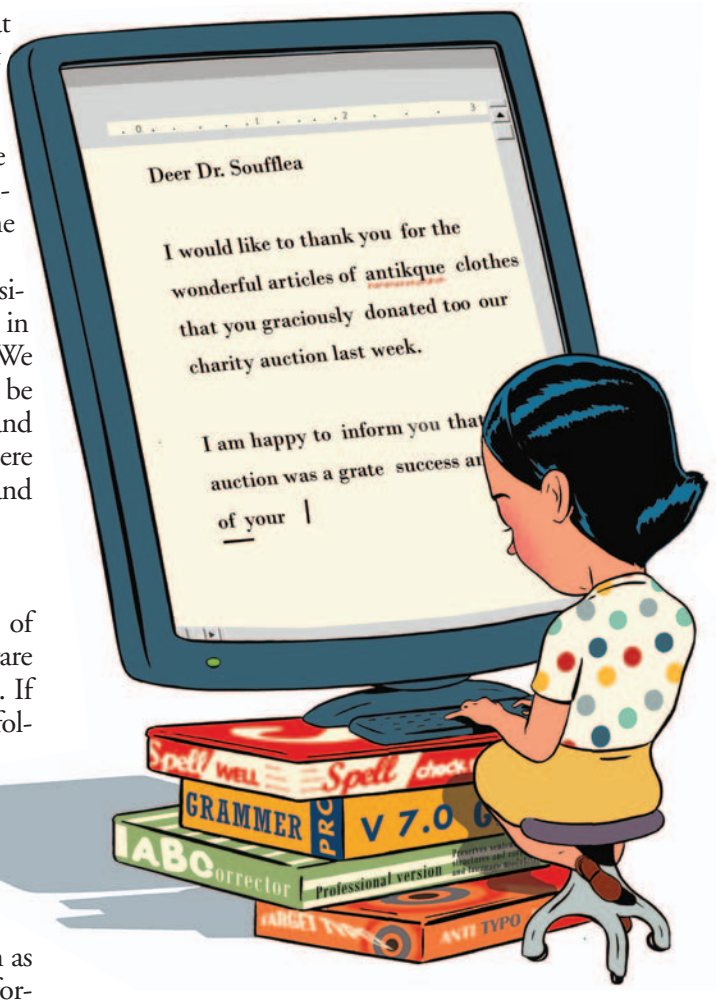
To uncover false negatives and overrule false positives, it is logical to expect that a user needs expertise in verbal skills to take full advantage of the software. We expected that users who are less skilled would be "fooled" by the imperfections in the spelling- and grammar-checking software. Our hunch was that there needed to be a fit between the task, the technology, and the person [4].

Prior Research

One area of research relevant to this study is that of computer credibility. Advice is given by the software and users must find it credible in order to follow it. If they do not find it credible, they will probably not follow it.

Credibility is composed of trustworthiness and expertise. It is sometimes equated with believability [3, 9], in that if a computer is deemed believable, it is considered credible. The notion that computers are credible is suggested by the use of terms such as "accepting the advice," "trusting," and "quality of information provided" [3].

According to Fogg and Tseng [3], research has shown many people view computers as not only "awesome thinking machines" [7], but "infallible sidekicks in the service of humanity" [3]. Over a decade ago, an account by Martin [7] tracked the credibility phenomenon to early depictions of computer technology by the popular press. The depictions sensationalized computer technology and failed to point out that computers perform only as they are instructed by their programmers and operators. The general public has not been informed that because humans sometimes make errors in those instructions, and complete testing of all possible interactions with the system is nearly impossible, computers are indeed prone to human errors.



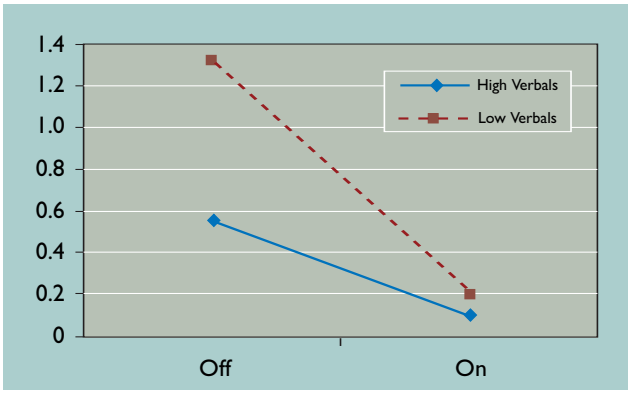


Figure 1. Uncorrected errors when flagged correctly.

Research in computer credibility has examined how credibility is gained, lost, and regained [5, 8]; how the context in which the computer is being used may affect credibility [5]; and how credibility is affected by individual user characteristics [1, 3].

Credibility is particularly important when computers are seen as decision aids; knowledge sources; and tutors or instructors [3]. This research considers computers as sources of knowledge and decision aids and therefore provides caution against overreliance on such software. Because of an increased level of credibility placed in the technology, users competent in the area in question may still permit technology to lead them to make incorrect decisions.

Research expectations. We expected to find that subjects with higher verbal ability would perform better on an editing task, and similarly, to find that subjects using the language-checking software would perform better. However, we expected that those of lower verbal ability would not receive the full benefit of the software. That is, they would not know when to ignore or overrule the advice.

Empirical Investigation

Following a pilot study of 20 participants, which enabled us to refine the experimental materials and task, we collected data from 65 participants (33 undergraduate and 32 graduate students at a major northeastern U.S. university). Standardized test scores (Scholastic Aptitude Test for undergraduate students and Graduate Management Admissions Test for graduate students) were obtained from school records with permission of the volunteers. Subjects above the median were labeled as "high verbals" and those below the median were labeled as "low verbals."

We asked participants to edit a business letter using Word 2003 with the language-checking software turned on for half the subjects, and turned off for the other half. Task performance was measured in terms of the three types of errors, of which five instances of each were incorporated in the text. The three types of errors were:

- Correctly identified errors;
- False positives (erroneous indications of an error); and
- False negatives (errors not flagged).

In general, we measured task performance by counting the number of remaining errors of each type in the document after the subjects tried to improve it. As in the game of golf, lower scores denote better performance. A score of zero represents a perfect score (finding and correcting all errors without falling for false positives).

Results

Some 28 subjects performed the task without the language-checking software, while 37 worked with the software on. The median for splitting undergraduate subjects into categories of high and low verbals was 570 (from a range of 460 to 720), and the median for graduate students was 34 (from a range of 21 to 44). As a result, 32 subjects were categorized as low verbals and 33 were high verbals. We examined the various demographics associated with each grouping and found that there were no unexpected significant differences among them that could confound the results obtained. It was particularly interesting that there was no significant difference between the graduate and undergraduate students in their performance (p-value = 0.33).

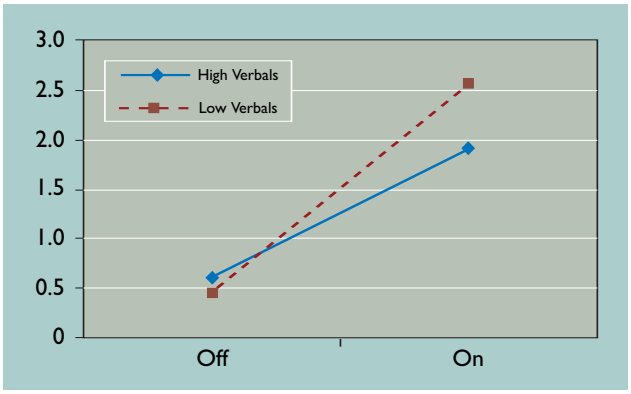


Figure 2. Uncorrected false positives.

Results will be presented for each error type separately. The first error type (errors correctly flagged) is expected to provide benefits to both groups, but to low verbals more than to high verbals because high verbals will be more likely to catch the problems by themselves. The second type (false positives) is expected to

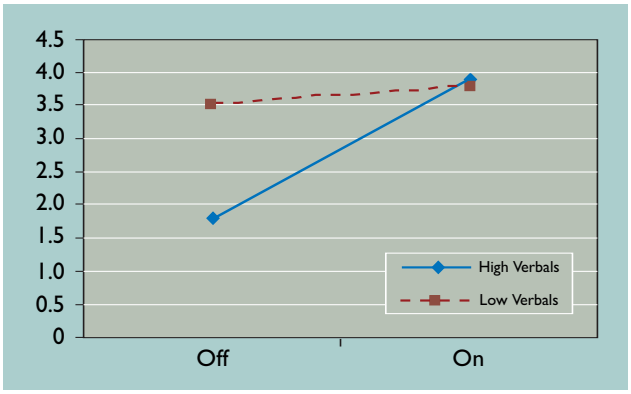


Figure 3. Uncorrected false negatives.

degrade the performance of only low verbals because those individuals are more likely to be fooled by the false indicator. The third type (false negatives) is also expected to degrade the performance, but more degradation is expected for low verbals than high verbals because low verbals are expected to place more faith in the unmarked text—items that failed to have been marked would then be missed.

Correctly flagged errors. Figure 1 provides the results in compact form. As expected, both groups have lower (better) scores when the language-checking software is on. Also as expected, low verbals performed much more poorly than high verbals, but improved their scores to a level indistinguishable from high verbals. This pattern matches what the software manufacturer designed the software to do. Statistical analysis revealed that both main effects and also the interaction effect were significant.

False positives paint a more interesting picture. Figure 2 shows high and low verbals both perform at about the same level when the language-checking software is turned off. With the software turned on, however, both groups leave more errors uncorrected. Statistical tests show the only significant effect is that of having the language-checking software on or off. Indeed, the software overshadowed any other effects, including that of verbal ability and any interaction between the two factors. Stated simply, high and low verbals alike fell for the false error messages, and both performed far worse. This effect held for undergraduate and graduate students alike; computer credibility seems alive and well in the 21st century.

False negatives are real errors not caught by the language-checking software. Figure 3 shows that high verbals performed much better at detection of those errors than low verbals with the software off. The best performance was demonstrated by high verbals without the software. However, when turning on the software, the false-negative detection performance of high verbals sinks to that of low verbals. Statistical analysis supports these conclusions, revealing strong effects of the language-checking software, verbal ability, and an interaction between the two. Surprisingly, high verbals left twice as many errors uncorrected when the software was on.

What does this mean? The results strongly suggest that high verbals count on language-checking software too much. They behave as if they are lazy and do not hunt for errors missed by the software. It is interesting that such errors do not show up on the screen, and therefore could remain "hidden" forever from writers unless they are corrected by others. If those problems are not made known to the writers they will not learn proper spelling and grammar over time.

Further, they will not learn about the credibility of spelling- and grammar-checkers without this feedback, either. Tseng and Fogg [9] speculated that if a computer reports the spelling to be correct in a document, and the user later finds that there is a misspelled word, credibility will suffer. Perhaps over time, as predicted by Martin [7], credibility expectations will become more realistic as users gain experience. Unfortunately, some related results are not very encouraging; error rates as high as 30% did not seem to destroy the credibility of an automobile navigation system [5].

It might be unreasonable to expect all users to form deserved levels of computer credibility. Zimmerman and colleagues [10, 11] found that when a TV show recommender system suggested programs already viewed by users, requiring little further effort, the users believed the software worked well and readily took its advice. On the other hand, when the system suggested unfamiliar programs, users believed the software had failed. The researchers concluded that users deem it unpleasant to find a system challenging them. Perhaps avoiding effort when the language-checking software does not find anything is pleasant, just like identifying only shows that the viewer already watches. In other words, users might be quite fervent in their hopes of avoiding more information processing steps and they more readily believe the document is free of errors when language software finds nothing. When the software finds and marks problems, they hastily tend to follow the software's suggestions.

Conclusion

In terms of overall performance, users were affected by both their level of expertise and the presence of the software that checked spelling and grammar. High verbals performed better than low verbals. For the most obvious errors properly discovered by the software, user performance was higher with the language-checking software turned on. However, considering false positives and false negatives—both common situations when editing a document—performance was worse for

both high and low verbals with the software turned on.

When errors are identified correctly by the language-checking software, leaving it on helps both high and low verbals eliminate errors. In this case, when the software is turned off, high verbals are not disadvantaged, while low verbals tend to be. This is the case the spelling- and grammar-checking software manufacturers expect, design for, and perhaps, hope for.

False positives and false negatives are much more revealing. With false positives (non-errors flagged as errors), both high and low verbals perform more poorly with the language-checking software on. Both groups are fooled by the software's incorrect advice, and ruin correct text by following that advice and making changes. It is possible the sampled subjects have a high degree of computer credibility, but because the effect was strong for both undergraduate and graduate students alike, we expect that such behavior is common to other groups as well.

Also interesting were the results found for false negatives (errors missed by the language-checking software). Performance on an editing task for both high and low verbals decreased when the software was turned on. When the language-checking software was on, high verbals tended to leave behind nearly twice as many mistakes than when it was off. This illustrates a possible false sense of security, perhaps due in part to credibility and in part to a preference to avoid effort and abdicate their responsibility to a computer agent that promises to do the cognitively demanding and tedious work. They attribute power and trust to the language-checking software rather than search the document for errors.

The level of trust that users attribute to spelling- and grammar-checking software, and perhaps also to a variety of intelligent agents, may not always be commensurate with the software's ability to do the job without error. Users must recognize they need to limit the extent of confidence and trust they place in such software when undertaking a document editing task. Hence, users should realize, even if unpleasant, that more of the onus is on them in preparing an error-free document. Rather than consider this a technology problem, it appears to be a behavior problem.

One way to reduce these problems might be to affix warning labels on office software, but the likelihood of that (possibly beneficial) scenario is quite low. We might need to rely on experience over a long period of time. When spelling and grammar checkers make obvious errors, perhaps it does us a favor because it puts us on alert that the software is fallible. However, when errors are not so obvious, we are likely to continue to be too trusting.

We originally conducted this study to warn those of lower verbal ability that spelling- and grammar-checking software might require additional verbal expertise. What we found surprising—perhaps more interesting, and more pragmatic—was that *all* users need such a label. Users with high verbal ability can actually lose their advantage and perform just like those with lower verbal ability. With or without a warning label, users should no longer abdicate their responsibilities to software that lacks the ability to analyze completely the meaning of their written words for them. It will remain a task more appropriately assigned to the human side of the equation for many years to come. C

References

1. Bauhs, J., and Cooke, N. Is knowing more really better? Effects of system development information in human-expert system interactions. In *CHI'04 Proceedings Companion.* (Boston, Apr. 24–28, 1994). ACM Press, NY, 99–100.
2. Connors, R.J. and Lunsford, A.A. Frequency of formal errors in current college writing, or Ma and Pa Kettle do research. *The St. Martin's Guide to Teaching Writing,* 2nd ed. Robert Connors and Cheryl Glenn, Eds. St. Martin's, New York, NY, 1992.
3. Fogg, B.J. and Hsiang, T. The elements of computer credibility. In *Proceedings of CHI99.* (Pittsburgh, PA, 1999), 80–87.
4. Goodhue, D.L. User evaluations of MIS success: What are we really measuring? In *Proceedings of the Hawaii International Conference on System Sciences.* (1992), 303–314.
5. Kantowitz, B.H., Hanowski, R.J., and Kantowitz, S.C. Driver acceptance of unreliable traffic information in familiar and unfamiliar settings. *Human Factors 39*, 2 (1997) 164–176.
6. Kies, D. Evaluating grammar checkers in modern English grammar; papyr.com/hypertextbooks/engl_126/gramchek.htm (2002)
7. Martin, C.D. The myth of the awesome thinking machine. *Commun. ACM 36,* 4 (Apr. 1993), 120–133.
8. Muir, B.M. and Moray, N. Trust in automation: Part II. Experimental studies of trust and human intervention in a process control simulation. *Ergonomics 39*, 3 (1996), 429–460.
9. Tseng, S. and Fogg, B.J. Credibility and computing technology. *Commun. ACM 42*, 5 (May 1999), 39–44.
10. Zimmerman, J., Kurapati, K., Buczak, A.L., Schaffer, D., Gutta, S. and Martino, J. TV personalization system. *Personalized Digital Television: Targeting Programs to Individual Viewers.* L. Ardissono, A. Kobsa, and M. Maybury, Eds. Springer, NY, 2004.
11. Zimmerman, J. and Kurapati, K. Exposing profiles to build trust in a recommender. In *Proceedings of CHI02* (Minneapolis, MN, Apr. 2002), 608–609.

Dennis F. Galletta (galletta@katz.pitt.edu) is a professor of business administration at the University of Pittsburgh and Temple University, Philadelphia, PA.
Alexandra Durcikova (alex@eller.arizona.edu) is an assistant professor of MIS at the University of Arizona.
Andrea Everard (everarda@lerner.udel.edu) is an assistant professor of MIS at the University of Delaware, Newark.
Brian M. Jones (bjones@tntech.edu) is an assistant professor of MIS at Tennessee Technological University, Cookeville, TN.