Approaches to Marking Electronic Texts

Martin Holmes
University of Victoria

A brief survey of the rather limited amount of research that has been done on the marking of writing shows that there remains a great deal of confusion with regard to the effectiveness of different marking strategies. Many second language teachers spend an inordinate amount of time marking students' texts (Zamel 80); according to Leki,

> Written comments are time consuming, but teachers continue to write comments on students' papers because we sense that our comments help writers improve; because written comments *seem* more feasible and more thorough than conferences on every paper; and because, for most writing teachers, our jobs require us not only to evaluate our students' writing but to be able to justify our evaluations (58).

Perhaps not surprisingly, students also expect and require detailed responses to their papers from instructors, noting all errors (Leki 62).

However, Cohen and Cavalcanti, in a study on teacher marking practices, found that "teachers' feedback ... is unclear, inaccurate and unbalanced ..." (155); Fathman and Whalley, citing Sommers, state that "... most teacher comments are vague and do not provide specific reactions to what students have written" (179); and Zamel notes that "... teachers' marks and comments usually take the form of abstract and vague prescriptions and directives that students find difficult to interpret" (79). While some evidence has been found that certain types of marking (specifically, drawing attention to errors rather than correcting them, and providing positive feedback rather than just negative prescription) *do* seem to improve student writing (Fathman and Whalley 187), this is contradicted by other studies which appear to demonstrate that no conceivable combination of praise, blame, commentary, diagnosis or correction "produced significant improvements

> "Written comments are time consuming, but teachers continue to write comments on students' papers because we sense that our comments help writers improve..."

in students' subsequent writing" (Leki 61).

In other words, it would seem to be the case that, although writing instructors dedicate a large proportion of their time to detailed and complex marking practices, and their students expect them to, their responses and comments are frequently less than competent, and may have no effect whatsoever on the quality of their students' writing. This is a rather disheartening state of affairs.

Recently, writing instructors have begun to deal with a new factor in their work: the electronic document. Students are increasingly submitting computer files (text documents, email or word-processor documents) instead of the conventional hard copy. In addition, more and more writing is taught by distance education through online systems, and here the electronic submission and return of documents may become the primary form of interaction between student and instructor. The techniques and methodology of marking assume a greater significance in this context.

It is not my purpose to examine what kinds of responses teachers should give to students' papers. Rather, this paper will look at four possible technical approaches to receiving, marking up, and returning a student text in electronic form, together with their strengths and weaknesses. Some of these methods can help to improve the clarity of the document returned to the student, distinguishing comments on form, for example, from comments on content; they can increase the consistency of an instructor's responses to particular errors or error-types; and they will enable us to return to the student a text which is more intact, in the sense of being less invaded or defaced by an overlay of multicolored inks and scrawled comments. The methods outlined here range from a simple text-based system, for which no special software or knowledge is needed, to more complex solutions involving macro programming and special software.

Method 1: A Text-Based Marking System

In this approach, the instructor simply inserts a number of symbols into the text, indicating error types. The student is provided with a key to the symbols, and is expected to correct the errors. Longer notes can be included using a simple numbered footnote system. There are many text-based marking systems in use, and instructors tend to evolve their own systems according to their needs. The system shown here is in use by the Comenius Group for a Business English course taught by the Internet: **http://www.comenius.com/writing/index.html.**

The Comenius Web site demonstrates how the

instructor actually uses a special toolbar and macros to automate the marking process, but this is not necessary in order to use this method of marking.

This key is taken (with permission) from the Comenius Web site:[1]

<u>CORRECTION SYMBOLS KEY</u>

][placed at the beginning of a long correction

[x] delete the word or words
    I want][for think[x] to go there.

[+] add the missing word or words
    Give me[+] the pencils.

[1],[2] ,...see the bottom of your assignment for a
        teacher comment
        I live in Tkyo[1] Japan.

[1] Be careful of spelling mistakes in business
    correspondence

[ ] add a blank line
    Sincerely,
    [ ]
    [ ]
    [ ]
    David Rogers

[#] mistake in number [singular/plural]
    I have three book[#].

[art] wrong or missing article
      This is [art] book.

[cap] capitalize or don't capitalize the word or
        words
        I visited the][united states of america[cap].
        I like Strawberry [cap] ice cream.

[prep] mistake or missing preposition
        He lives at [prep] Madison Avenue.

[ww] wrong word
      I except[ww] your advice.

[wf] wrong word form
   He talked about go [wf] there.

[tns] mistake in verb tense
   I eating[tns] everyday at 8:00.

[pro] missing pronoun
   [pro] is dangerous to travel at night.

[frag] sentence fragment
   Since you are so nice [frag].

[r/on] run on
   I like my friend she makes me laugh [r/on].

[sp] spelling mistake
   I practice[sp] English every day. [sic]

"Among the advantages of this method are the fact that it is technologically simple; student texts can be submitted and returned by e-mail, without the need to ensure that instructor and students are using the same platform or the same software."

Among the advantages of this method are the fact that it is technologically simple; student texts can be submitted and returned by email, without the need to ensure that instructor and students are using the same platform or the same software. In addition, it is relatively easy for a student to edit or redraft an essay marked in this way, working directly on the marked text. Finally, this approach (along with the other three outlined in this article) allow the instructor to insert comments of any length. In traditional marking, the instructor is often constrained by the physical space available in the margins of a paper, and as a result comments are often inscribed in tiny handwriting, or reluctantly truncated. In all of the methods shown here, the instructor has unlimited space, because the electronic document expands as text is inserted.

However, this approach produces output that is less than satisfactory in a number of respects. In a simple text document, the instructor's comments and annotations are not distinguished from the student's original text except (in the example above) by being enclosed in square brackets. In a similar traditional marking system, pen color would be used to distinguish instructor's comments from the student's own writing. The textbased approach does not allow for this.

Method 2: Word-Processor Revision Tools

"[T]he prevalence of one error-type is immediately obvious because of its color, and error diagnoses (marked by highlighting) are clearly distinct from comments on content or structure (which would be footnotes)."

Most modern word-processing programs include a range of tools intended to be used for reviewing and revising documents. This system uses the colored highlighter, a revision tool available in recent versions of some word-processing programs. In this example, six basic colors are used as codes for simple error types. In addition, footnotes can be used to add longer comments where required.

These are the basic colors and the error-types they signify:

Yellow	Spelling
Grey	Punctuation/Capitalization
Red	Verb error (tense or agreement)
Blue	Noun error (number or form)
Purple	Other grammatical error
Green	Word choice error

In actual fact, word-processors usually allow a much wider range of colors to be used. However, in practice, colors which are not members of the basic system palette may not display correctly on a variety of platforms or in different applications (see below), and the system itself becomes difficult to apply and interpret easily when larger numbers of colors are used, since the colors (unlike the abbreviations in the previous method) carry no mnemonic value.

The following additional revision and text-formatting features can be used to make this system more flexible:
1. Underline and double-underline
2. Bold
3. Italic
4. Superscript and subscript
5. Font colors
6. Strikethrough (indicating text to be deleted)

The instructor would, of course, have to ensure that any of these textual features used for marking were not also used by the student for other purposes within the document.

This method has some advantages over text-based marking. The instructor's annotations are more easily distinguished from the original text. In addition, the prevalence of one error-type is immediately obvious because of its color, and error diagnoses (marked by highlighting) are clearly distinct from comments on content or structure (which would be footnotes).

However, a new set of limitations arises when we use these kinds of tools. Only the more recent versions of major word-processing programs support colored highlighting,

so both the student and the instructor must be using compatible software. If they are not using the same application, a file format must be agreed upon to allow files to be exchanged. The logical choice here is Rich Text Format, which is intended to be a platform- and application-independent, ASCII-based document format. There are, however, many incompatibilities in the interpretation and display of textual features in RTF in different applications (see my 1996 article "Rich Text Format Survey").

In addition to the file format problem, the fact that word-processor files (rather than just text) must be exchanged introduces a new level of complexity into the marking process. Students must now save their work in a particular format, and (if submitting electronically) attach this file to an email message. The instructor must go through a similar process to send the document back to the student, who must be able to retrieve and open an attachment. Some learner training (and possibly instructor-training too) will probably be required.

A final objection to this kind of system is that it is extremely time-consuming. It is difficult to remember the color codes, and tiresome to change the highlighter color each time a new error-type is encountered. Impatience with this approach led to my developing a more efficient system for marking within a word-processor, based on macros and a special toolbar.

Method 3: Using a Toolbar with Macros

A macro is a small "program within a program", that is, a script which performs a sequence of actions within an application. Most word-processors, spreadsheets and other productivity software allow users to script or record macros to automate repetitive tasks. Macros can then be attached to buttons on the toolbar of the application, so that clicking on the button runs the macro, which performs the series of operations automatically. Macros can be used to automate the insertion of complex annotations into the text of a document, as shown in Figure 1 (see next page).

Here, when the instructor encounters an error (such as a spelling mistake), she selects the relevant text, then presses the button labelled "SP". This runs a macro which performs the following actions:
1. Double-underlines the selected text.
2. Turns that text red.
3. Inserts a blue superscript annotation "Sp".

Buttons for a range of error types are included on the

toolbar, as well as buttons for inserting a green superscript "Good" into the text, for inserting footnotes in green, and for changing selected text to different colors.

Student documents can be received as word-processor or text files, or as email, in this system. However, since enhanced textual features are used in the marking process,

Figure 1.

some kind of document file must be returned to the student. There are four options:

1. Native file format: If the student is using the same word-processor, the document can be saved in the native format of the program.

2. Rich Text Format: RTF (discussed above) will preserve the textual features used here. The file can be attached to an email message.

3. HTML format: Most major word processors will now save files in HTML format (as WWW pages). The file can then be opened in any Web browser.

4. Printed format: If necessary, the file can simply be printed out for the student.

The first three options require that the file be attached to an email message or returned to the student on a disk.

This approach to electronic marking (first outlined in my article "Marking Student Work on the Computer") has a number of advantages over those examined above. First, through the use of red, blue and green font colors (not visible in the illustration here) the instructor's comments and error diagnoses are clearly distinguished from each other, and from the student text. In addition, the use of double-underline and superscript ensures that even if this

document is printed on a monochrome printer, these distinctions will be preserved. The original formatting of the student's document (including indents, bold and italic text and so on) can be preserved. Finally, this system is considerably faster than the first two methods, because the macros automate the process of inserting annotations.

However, the text returned to the student remains a static document. Unless the student is required to revise or edit the text to produce another draft, then there is little to compel her to examine it, and no way to interact with it. The fourth approach described below is an attempt to remedy this.

Method 4: A Special Marking Program

In April 1996, Roy Bowers posted on the World Wide Web a demonstration of the use of hypertext to mark a student text in such a way that the student would be able to interact with it (Bowers 1996). His example uses three frames. In the largest is the student text, which contains links. Clicking on a link causes a comment or error diagnosis to be displayed in one of the other frames. If the user requires more information (such as a full grammar explanation), then clicking on a link in the second frame will display a more detailed explanation in the third.

I was intrigued by this approach, but I could see two major objections to it. First, I was not happy with the use of frames, which at that time were not supported by many browsers. Also, using frames requires multiple documents; the student would be required to receive at least four files, and load the correct one into the browser, for the system to work. Secondly, Bowers had (as far as I know) no system for creating these pages efficiently; essentially, each had to be coded by hand. My marking program, *Markin*, is an attempt to overcome these difficulties.

The interface of the program works very much like the word-processor macro toolbar described above, and it will also output the marked essay in the form of a Rich Text Format document which can be viewed in a word-processor, using coloured superscript annotations and underlining to distinguish the marking from the original text. However, the program will also produce output in the form of a Web page. The student's original text is intact at the beginning of the page. The text contains links, which jump to diagnoses or comments further down the page, as shown in Figure 2. In contrast to Roy Bowers's frame-based approach, this uses a single Web page, which means that only one document needs to be sent to the student.

Here, the subject-verb agreement error ("my parents *was...*") becomes a link. When the student clicks on the link, the browser jumps to another part of the document where the error-type is displayed. Comments can be inserted in the same way, and more general feedback can be appended to the essay in colored text, along with a grade. The program automatically compiles a list of each error type and the number of times it has been made, so problem areas are immediately apparent. This gives the instructor a simple way of tracking a student's progress through multiple assignments. The program itself is not unlike a simple word-processor:

An Example Marked Student Essay:
My Early Childhood

I was born in 1975 in Taiwan. When I <u>was small</u> child, my parents <u>was</u> very poor, so we did not have good food to eat or a nice house to <u>living in</u>. My father was a high school teacher of <u>Mathematics, my</u> mother was a secretary. They <u>work</u> very hard to earn enough money to <u>give</u> food for me and my two sister.

Error type:

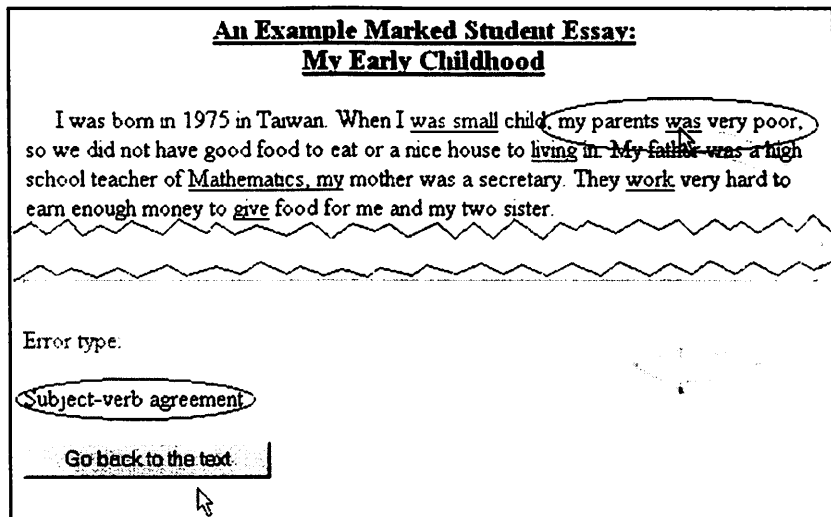Subject-verb agreement

Go back to the text.

Figure 2.

The text is marked using a toolbar of buttons similar to that used in Method 3, but this time the buttons can be quickly and easily customized to suit the level of the student and the marking preferences of the instructor. The program compiles the marked essay into a Web page automatically, and as well as the RTF and HTML output, it will also generate a text-only version of the page; this can be used if the student is not able to receive attachments.

When students interact with a document like this, instead of seeing a correction or a diagnosis directly, a two-stage process takes place. First, the presence of a link alerts the student to the fact that something is wrong. She may try to do the diagnosis herself. Then, she may click on the link to confirm her conclusion. The student text itself remains relatively intact and readable (it is not defaced by intrusive annotations) and moving between the text and the

associated annotations and buttons is simply a matter of clicking with the mouse. Besides error diagnoses, three other types of response can be provided for: praise annotations (using the bottom row of buttons), comments (linked to the text), and global feedback at the end of the essay, which can be coded in different colors.

Many other features have been added to the program since its initial development, to enable more rapid and consistent marking. The instructor can maintain a database of "useful text" snippets, which can be inserted into comments or feedback by selecting them from a list. This obviates the need to type out an explanation of a common point or issue repeatedly. Similarly, the instructor may build up a "useful links" database, containing links to useful Web pages; these references can be inserted wherever necessary, so it becomes easy to refer a student to a more detailed explanation of a particular point from the rapidly-expanding range of such resources available on the World Wide Web. Finally, the program is able to compile error statistics from multiple pieces of work, and output the results to a spreadsheet, so it is possible to do quite complex analyses of the work of particular groups of students, or of one student over a period of time.
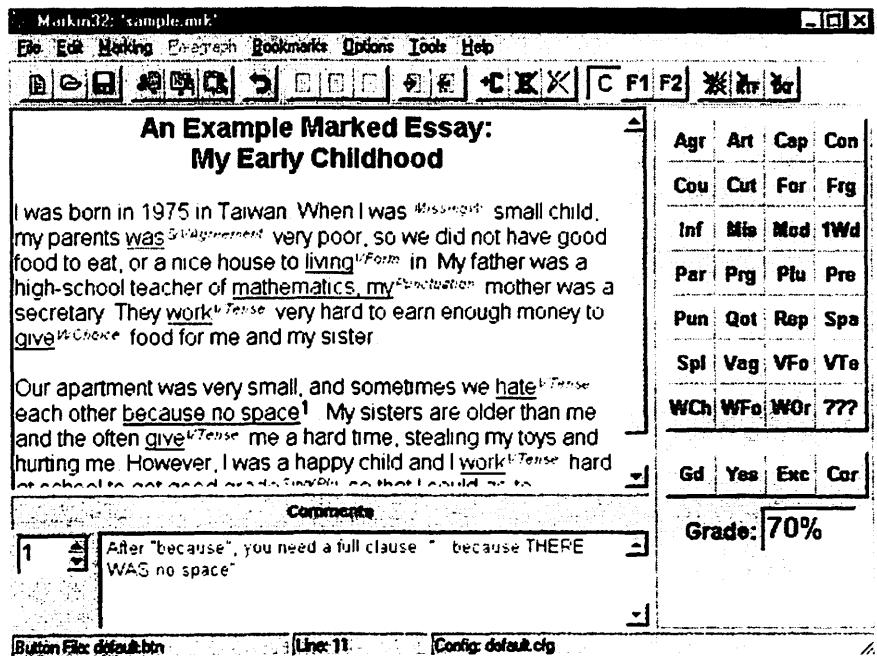
> "This approach has been in use at the University of Victoria for two semesters, and has proved very popular with both instructors and students."

Figure 3
Markin (called *Markin32* in its most recent versions)

This approach has been in use at the University of Victoria for several years, and has proved very popular with both instructors and students. However, there are some drawbacks to using a system like this. Because of the limits of hypertext, the same portion of student text cannot be linked to two different responses. Also, the student cannot directly edit the HTML output itself to produce another draft (although she can edit the RTF or text-only versions); to overcome this, with my own students, I usually send back both an HTML version and an RTF version of each marked piece of work. Finally, the student has to be trained to receive an HTML file by e-mail and open it in a browser, and the instructor needs to master the use of the *Markin* program.

Conclusion

While none of the approaches described above is a perfect solution to the marking problem, all of them offer some improvement over traditional pen-on-paper marking. As education increasingly goes online, and marking electronic texts becomes part of the daily work of an increasing number of instructors, institutions will need to begin evaluating solutions such as these to develop systems and policies which enable the rapid return of electronic documents marked in a clear and helpful fashion. Ultimately, it is likely that large software companies will begin producing expensive commercial products aiming to fill this need. Meanwhile, most of the groundwork is being done by instructors themselves, and the more ideas generated and tested during this stage, the better able we will be to evaluate, and even to influence the development of, such commercial products.[2]◆

Works Cited

Bowers, Roy. "Using HTML for online editing". Apr 22, 1996. http://www.tnis.net/rbowers/demo.html. (Aug 11 1997; unavailable July 5, 1999).

Cardelle, Maria and Lyn Corno. "Effects on Second Language Learning of Variations in Written Feedback on Homework Assignments." *TESOL Quarterly* 15.3 (1981): 251-261.

Cohen, Andrew and Marilda Cavalcanti. "Feedback on compositions: teacher and student verbal reports." Kroll 155-177.

Dillon, David and Dennis Searle. "The Message of Marking: Teacher Written Responses to Student Writing at Intermediate Grade Levels." *Research in the Teaching of English* 14.3 (1980): 233-242.

Fathman, Ann and Elizabeth Whalley. "Teacher response

to student writing: focus on form versus content." Kroll 178-190.

Holmes, Martin. "Rich Text Format Survey." May 25, 1996. http://web.uvic.ca/langcen/rnd/marking/richtext.htm. (July 5, 1999).

"Marking Student Work on the Computer." *The Internet TESL Journal*. Sep. 1996. http://www.aitech.ac.jp/~iteslj/Articles/Holmes-ComputerMarking/index.html. (July 5, 1999).

Kroll, Barbara, ed. *Second Language Writing: Research insights for the classroom*. Cambridge: Cambridge University Press, 1990.

Leki, Ilona. "Coaching from the margins: issues in written response." Kroll 57-68.

Robb, Thomas, Steven Ross and Ian Shortreed. "Salience of Feedback on Error and Its Effect on EFL Writing Quality." *TESOL Quarterly* 20.1 (1986): 83-93.

Semke, Harriet. "Effects of the Red Pen." *Foreign Language Annals* 17.3 (1984): 195202.

Sommers, N. "Responding to Student Writing." *College Composition and Communication* 33 (1982): 148-156.

Zamel, Vivian. "Responding to Student Writing." *TESOL Quarterly* 19.1 (1985): 79101.

Notes

[1]Last Modified: Thursday, September 26, 1996. Copyright 1995-1997, The Comenius Group, comenius@interport.net.

[2]The program *Markin* and macro/toolbar packages for *WordPerfect 6.1* and above and *MS Word 95* are available for download from: http://www.net-shopper.co.uk/creative/education/languages/martin.

An independent review of a new Windows 9x/NT version of *Markin*, *Markin32*, was published by *The CALICO Review* in May 1998, and can be seen at: http://www.arts.monash.edu.au/others/calico/review/markin.htm.

Martin Holmes is currently a programmer/consultant in the University of Victoria Language Centre. He holds a BA in English, an M.Phil, and the RSA DipTEFLA. He previously taught EFL for 15 years in several countries, and now provides Windows and Web-based programming services to support university language teaching and research.