

Genre-specific Error Detection with Multimodal Feedback

RELC Journal
2020, Vol. 51(1) 179–187
© The Author(s) 2020
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/0033688219898282
journals.sagepub.com/home/rel



John Blake 
The University of Aizu, Japan

Abstract

A purpose-built online error detection tool was developed to provide genre-specific corpus-based feedback on errors occurring in draft research articles and graduation theses. The primary envisaged users were computer science majors studying at a public university in Japan. This article discusses the development and evaluation of this interactive, multimodal tool. An in-house learner corpus of graduation theses was annotated for errors that affect the accuracy, brevity, clarity, objectivity and formality of scientific research writing. Software was developed to identify the errors discovered and provide learners with actionable advice and multimodal explanations in both English and Japanese. Qualitative evaluation received in usability studies and focus groups from both teachers and students was extremely positive. Preliminary quantitative evaluation of the effectiveness of the error detector was conducted. Through this pedagogic tool, learners can receive immediate actionable feedback on potential errors, and their teachers no longer feel obliged to check for common genre-specific errors.

Keywords

Error detection, technology-enhanced learning, corpus-based, research writing, learner corpus

Introduction

Teaching Context

This article discusses the development and evaluation of a purpose-built online error detector that provides multimodal feedback. This interactive tool was developed to provide genre-specific corpus-based feedback on errors occurring in draft research articles and graduation theses. The primary envisaged users were computer science majors studying at a public university in Japan. This university requires all students to submit graduation theses written in English. The university specializes in computer science and so the thesis for

Corresponding author:

John Blake, The University of Aizu, Aizuwakamatsu, 965-8580, Japan.
Email: jblake@u-aizu.ac.jp

undergraduates takes the form of a short research article. The format and style of the article replicates computer science journals. The thesis serves as a vehicle for students to learn rather than to contribute to the research literature. Undergraduate students face two language-related problems when writing their thesis: (1) lack of proficiency in written English, and (2) lack of familiarity with the genre of scientific articles. To address these problems, a thesis writing course is offered in the final semester of their senior year. Undergraduates registered on this course are required to submit sections of their thesis to their writing tutor for comments. These submissions are permeated with surface-level mistakes.

Reason for Innovation

Aside from the desire to help teachers and learners, three other reasons underpin the development of this error detector: pedagogic motivation, genre specificity and multimodality.

Pedagogic Motivation. In general, error detection tools aim to locate and automatically correct errors. Given the context-specific nature of many errors, the correction that is suggested may not be suitable. Moreover, learners need to engage with the feedback to improve as writers (Zhang and Hyland, 2018). This detector not only identifies but also categorizes and explains errors. Engagement is encouraged by providing knowledge about language (Fernández-Toro and Hurd, 2014), enabling learners to make informed choices. Students with limited proficiency in English tend to accept automated grammar-check suggestions with little thought, resulting in a reliance on the technology, a lack of cognitive engagement and negligible learning. For example, the grammar check function in Microsoft Word identifies many errors but there is a lack of description and explanation. Learners may simply accept the suggestions without understanding them. By providing sufficient information to understand their errors, learners are armed with data on which to make their own decisions. Unlike generic error detection tools that identify errors and suggest replacements, this tool aims to help the learner understand the reasons for the errors and, where appropriate, allows learners to make choices on how to revise the wording.

Genre Specificity. For specific genres with high generic integrity (Bhatia, 1996) such as research articles, it is possible to pinpoint errors more easily and more accurately. The following example shows how error detection can be improved by ruling out particular phraseologies. Sentence 1 was one of the many sentences discovered in the learner corpus that incorrectly used the expression ‘*There happened*’.

- (1) *There happened a problem in the software.
- (2) There happened to be a problem in the software.

Sentence 1 is ungrammatical because *happen* is an intransitive verb and so should not be followed by a direct object. At the time of writing, Grammarly, a generic error detector, is unable to detect this error. To discover this error, a simple search for ‘*There happened*’ could be used. However, sentence 2 contains the same string, or sequence, of two words; but in this case, the sentence is grammatically accurate. Developers of automated error detection systems strive to avoid false positive errors. This is in line with the finding by

Table 1. Criteria for Graduation Theses.

Type of Error	Description
Accuracy	Factual and language errors
Brevity	Using too many words
Clarity	Using vague or ambiguous terms
Objectivity	Using terms that appear subjective
Formality	Using abbreviations, contractions, and informal terms

Source: In-house thesis writing course.

Nagata and Nakatani (2010) that inaccurate feedback is worse than no feedback. However, given that in the learner corpus, there were multiple instances of ‘*There happened*’ used inappropriately but no instances of accurate use; it seems reasonable to include it in an error detector targeted at Japanese computer science majors.

Multimodality. As student preferences for feedback styles, language, and medium of explanation vary greatly (e.g. Ferris et al., 2013); explanations are offered in different modalities (text, audio and video) and languages (English and Japanese).

Description of the Innovation

This tool is tailored to address the specific needs of one set of learners, namely Japanese learners with limited proficiency in written English who draft short research articles in the field of computer science. Generic error detectors are unable to meet the needs of these students. The genre-specific nature of this niche error detector enables detection of errors that other more sophisticated detection tools overlook.

This error detector is designed to enable novice writers to conform to the generic expectations of computer science research articles, specifically in terms of accuracy, brevity clarity, objectivity and formality. These five categories of errors are used extensively in the thesis writing course (see Kaneko et al., 2018 for a detailed description). Table 1 shows novice writers the criteria used to evaluate the language in their graduation theses. Learners are encouraged to systematically review their writing focussing on one criterion at a time. This error detector, therefore, dovetails into the editing stage of thesis writing. A notable aspect is the pedagogic design that provides actionable advice and makes extensive use of audio and video to provide additional explanations on demand. This online tool is designed to be used both in and out of the classroom, making it suitable for distance, blended and flipped learning.

Development

Overview

To create this tool, commonly-occurring errors were extracted from a specially-created corpus of graduation theses annotated for the five types of errors found in scientific

writing. Software was then developed to identify potential errors. Feedback materials in text, audio and video formats were designed and developed to provide actionable advice and enable learners to reveal the multimodal explanations on demand. The three phases of corpus, software and feedback development are described below.

Corpus Development Phase

A learner corpus of all graduation theses submitted over a three-year period (2014–2016) was compiled. From this corpus of 629 theses, batches of ten theses were selected for annotation. Errors were manually and automatically identified and classified into five types using the UAM Tool (O'Donnell, 2008). By the 50th thesis, few new errors were being detected: saturation had been reached and annotation was discontinued. The errors were extracted into an error bank. Harnessing a tool from quality management, namely a failure modes and effects analysis (Gilchrist, 1993; Stamatis, 2003), errors were assigned values for frequency, severity and detectability. The weighted score for each error was calculated. A full description of the corpus specification, development, annotation and analysis is provided in an earlier work (Blake, 2018).

Software Development Phase

Coding started with errors with the highest weighted priority; but given the necessity to balance cost and performance, not every error was included. For each error that was included, an alphanumeric code was assigned and a regular expression was created to parse for the error. Regular expressions are used to search text and match particular permutations of characters, letters or words (see Friedl, 2006; Watt, 2005 for detailed descriptions). This is akin to the way that the autocorrect function in Microsoft Word automatically notices the typo *htis* and replaces it with *this*.

Figure 1 shows the web interface that was created to enable learners to submit their drafts online. There is a one-minute tutorial video to show less computer-savvy learners how to operate it. The function of each toggle button is displayed as the cursor hovers over the button. Once learners paste their text into the submission box, the output is immediately generated.

The following short paragraph was written by a student to explain the importance of a research topic.

This researchs focuses on improving the system design. I think that this will help many students learn programming more good than the current system. You will be able to learn programming quickly. It's very effective.

Figure 2 shows the output of the error detector for this paragraph. The accuracy check discovers the word *researchs* and suggests using the uncountable noun *research* or the countable noun *studies*. The brevity check detects and recommends deleting the unnecessary sentence stem *I think*. The clarity check identifies ambiguity in *good* and suggests using a more specific term or providing a definition. The objectivity check finds the pronoun *you* and suggests replacing it with a more specific noun or using passive voice.

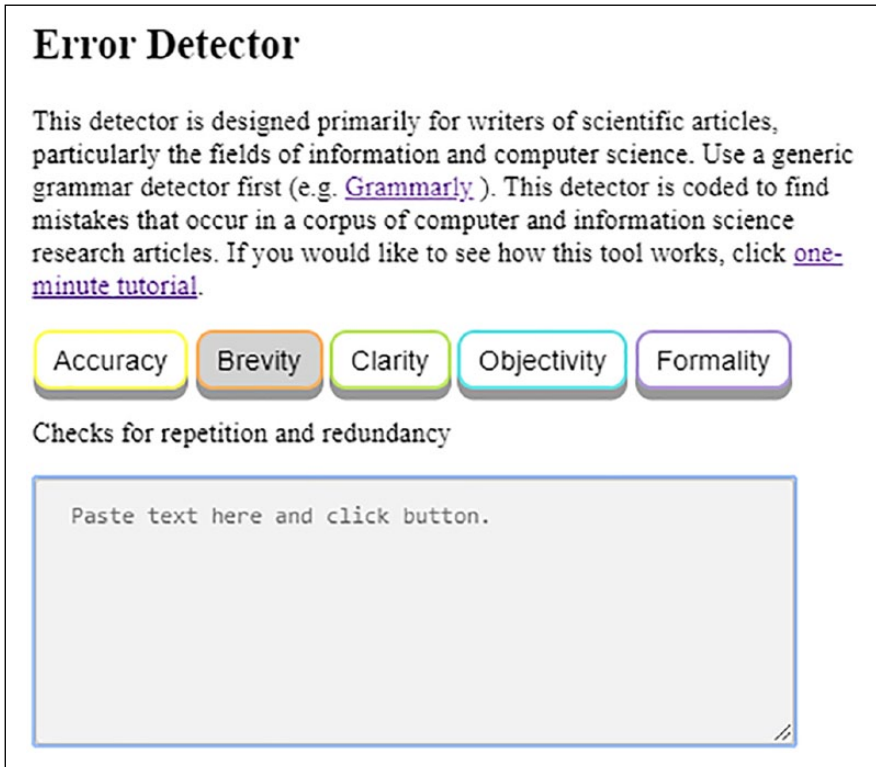


Figure 1. Error detector web-based interface.

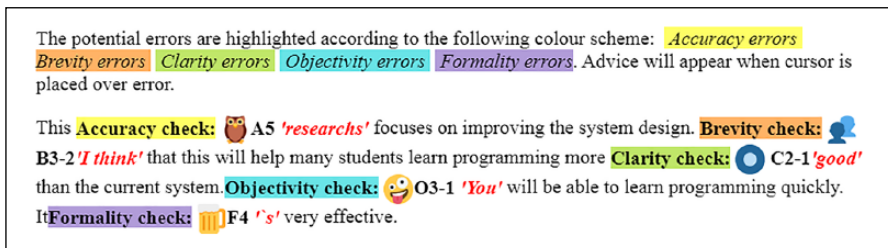


Figure 2. Output of error detector, showing error type, code and emoticon.

Finally, the formality function identifies that the apostrophe is from the wrong character set. It is worth noting that although Microsoft Word can identify errors with *researchs* and *more good*, many computer science majors write their theses in LaTeX (Lamport, 1994). LaTeX uses plain text so students need to add code in angled brackets to format their thesis unlike in formatted text systems, such as Microsoft Word. By default, there is no spell or grammar check facility either.

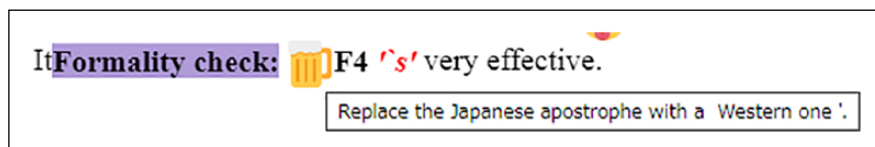


Figure 3. Pop-up comment revealed on hover.

Learners copy and paste their text into the submission box and feedback is automatically generated. Once an error is identified, an emoticon is displayed to show the error type, an alphanumeric code is given to pinpoint the specific type of error within the category, and a keyword from the submitted text is colorized. Users can click on any of the five toggle buttons to show or hide different types of errors. Advice can be revealed by moving a cursor over the feedback (see Figure. 3), and audio or video explanations can be accessed.

Feedback Materials Creation Phase

Although initial releases of earlier versions of this error detector were well-received, learners were reluctant to read textual explanations. Learners have come to expect online learning resources to be interactive, highly visual and multimodal (Hafner et al., 2015), and so audio and video explanations were incorporated. To increase the pedagogic effectiveness of the explanations, extraneous processing of spoken and written text was minimized by reducing redundancy and placing visuals close to corresponding text to increase spatial contiguity (Mayer, 2009). Reductions to the reading burden and the cognitive load were achieved by minimizing text, harnessing emoticons and using colour effectively. A slideshow video format was selected accompanied by voice over. This enabled video explanations to be created in two stages (slides and audio) and then combined.

Evaluation

Evaluation was conducted throughout the development of the error detector using usability studies in which small cohorts of learners tested the detector. They were observed and asked for feedback on various aspects of the tool. Preliminary quantitative evaluation was conducted by counting the instances of detected errors. Qualitative evaluation consisted of direct individual and focus group feedback. Each of these is described in turn.

Usability Studies

Usability tests were conducted regularly to identify aspects to improve. This error detector has been under development since 2012 and is now in its 50th version. The initial version comprised a single error category with feedback provided inline. Refinements were made based on usability studies and feedback received from users directly and focus group interviews. Some learners wanted to be able to search for one type of error while others preferred to search for multiple types simultaneously. To allow this, the five

Table 2. Number of Errors Detected in Student Submissions of Draft Theses.

Student Number	Number of Errors Detected	
	Initial submission	Final submission
1	12	10
2	-	0
3	-	25
4	19	8
5	-	26
6	23	20
7	8	5
8	-	16
9	0	0
10	38	17

Table 3. Number of Errors Detected in Each Submission of Student 10.

Submission number	1	2	3	4	5	6	7	8	9
Number of errors detected	38	38	39	31	16	17	16	15	17

toggle buttons already described were added. Emoticons were added to the error detector to reduce the reading load. Alphanumeric codes were included to enable explanations to be linked to the errors more easily. Initially feedback messages were given inline, but learners commented that this was intrusive and as they frequently made the same error multiple times, they did not need to read the comment on each instance. Inline comments were replaced by pop-up comments that can be revealed on demand. In the latest usability study, learners described the error detector as easy to use.

Quantitative Evaluation

The error detector was introduced to ten final-year students enrolled in the elective thesis writing course. Students were advised to use a generic error check, such as Grammarly, first and then use the genre-specific error detector. During this one-semester course, students submitted up to nine drafts of their thesis. The initial and final submissions were submitted by the class tutor into the detector. The number of detected errors in those versions are shown in Table 2. As can be seen, some students failed to submit their initial version. Students who submitted theses that had zero errors detected were asked about their drafting process. All stated that they had already submitted their work into Grammarly and the genre-specific error detector and acted on the suggestions received. Student 10 worked closely with the thesis writing tutor and submitted nine drafts for comments. Table 3 shows the number of errors detected in each version. In total, 227 errors were detected. The first four submissions focussed on developing the introduction, method, results and discussion, adding approximately 500 words to each successive

draft. The final five submissions included revisions based on language feedback from the thesis writing course tutor and feedback on content from the thesis supervisor.

Qualitative Evaluation

Positive qualitative evaluations were received in usability studies and from focus groups comprising teachers and students. A focus group discussion with teachers noted that the error detector finds many of the errors that commonly occur in graduation theses. Comments from teachers tended to focus on the increased readability of texts that students submit after having used the error detector. Students' comments were overwhelmingly positive. Many students regularly used the error detector when drafting. Some students preferred to see all five types of errors visualized at the same time while others preferred to work on one or two types of errors at a time. The majority of students stated that initially in the early stages of drafting, errors in all five categories occurred, but in the later stages errors in the accuracy category predominated. Although both quantitative and qualitative evaluations were positive, both teachers and students commented that further fine-tuning is necessary.

Reflection

With this genre-specific error detector, learners can receive actionable feedback instantly - at anytime and anywhere. The multimodal bilingual explanations allow them to select their preferred mode and medium, addressing learner preferences. The explanations, in turn, enable learners to understand more about the errors identified. Teachers who teach scientific, technical or academic writing to computer science majors no longer need to check for the surface-level genre-specific mistakes that can be automatically identified. This frees up the teachers' time, enabling them to focus on deeper or more complex errors. This automated error detector can save a substantial amount of time when dealing with long texts or large classes. One important take-away for teachers and materials developers alike is the need to write unambiguous actionable advice. A key challenge in this project was writing advice that is specific enough to resolve the detected error, yet general enough to apply to other related problems without falling victim to overgeneralization.


Future Pedagogical Directions

This detector aims to address genre-specific surface-level errors occurring in graduation theses of computer science students. This category is rather fuzzy and so, at times, overlaps occur with more generic grammatical error detectors. Interactive teaching materials will also be developed from the bank of errors, explanatory audio and video files. These materials will be closely integrated into the online thesis writing course. The trend in error detection is to harness deep learning and big data, but perhaps more targeted corpora and automated error detection focussed on specific niches would be more appropriate for language learner needs. The latest version of the error detector is available at: <https://jb11.org/error_detector.html>.

Funding

The author received no financial support for the research, authorship, and/or publication of this article.

ORCID iD

John Blake  <https://orcid.org/0000-0002-3150-4995>

References

- Bhatia VK (1996) Methodological issues in genre analysis. *Hermes* 16: 39–60.
- Blake J (2018) Corpus-based error detector for Computer Science. In Y. Tono and H. Isahara (eds.). *Proceedings of the Fourth Asia Pacific Corpus Linguistics Conference*, (pp. 50–54). Takamatsu, Kagawa, Japan: Asia Pacific Corpus Linguistics Association.
- Fernández-Toro M, Hurd S (2014) A model of factors affecting independent learners' engagement with feedback on language learning tasks. *Distance Education* 35(1): 106–25.
- Ferris DR, Liu H, Sinha A, and Senna M (2013) Written corrective feedback for individual L2 writers. *Journal of Second Language Writing* 22(3): 307–29.
- Friedl JE (2006) *Mastering Regular Expressions*. Sebastopol, CA: O'Reilly Media.
- Gilchrist W (1993) Modelling failure modes and effects analysis. *International Journal of Quality and Reliability Management* 10(5): 16–23.
- Hafner CA, Chik A, and Jones RH (2015) Digital literacies and language learning. *Language Learning and Technology* 19(3): 1–7.
- Kaneko, E., Park, M., Wilson, I., Heo, Y., Roy, D., Yasuta, T., Nicholas, A. and Blake, J. (2018). English Curriculum Innovation for Computer Science Majors in the Japanese EFL Context: From Needs to Tasks. *Proceedings of 2018 IEEE ProComm Annual Conference* (pp. 84–89). Toronto, ON, Canada: IEEE. Available at: <https://doi.org/10.1109/ProComm.2018.00028>.
- Lampport L (1994) *LATEX: a Document Preparation System: User's Guide and Reference Manual*. Reading, MA: Addison-Wesley.
- Mayers RE (2009) *Multimedia Learning (2nd edition)*. Cambridge: Cambridge University Press.
- Nagata R, Nakatani K (2010). Evaluating performance of grammatical error detection to maximize learning effect. In: Association for Computational Linguistics, *Proceedings of the 23rd International Conference on Computational Linguistics: Posters, 894–900*. Stroudsburg, PA: Association for Computational Linguistics.
- O'Donnell M (2008) The UAM corpus tool: software for corpus annotation and exploration. In: *Proceedings of the XXVI Congreso de AESLA, 3–5*. Almeria Spain: The Spanish Society for Applied Linguistics (AESLA).
- Stamatis DH (2003) *Failure Mode and Effect Analysis: FMEA from Theory to Execution*. Milwaukee, WI: ASQ Quality Press.
- Watt A (2005) *Beginning Regular Expressions*. Indianapolis, IN: Wiley Publishing.
- Zhang ZV, Hyland K (2018) Student engagement with teacher and automated feedback on L2 writing. *Assessing Writing* 36: 90–102.