Universal Design 2024: Shaping a Sustainable, Equitable and Resilient Future for All
K.S. Fuglerud et al. (Eds.)

doi:10.3233/SHTI240998

Should Student Coursework Be Universally Designed? Universal Design Requirements May Prevent Academic Cheating

Frode Eika SANDNES[a], Morten TOLLEFSEN[b], Jo HERSTAD[c], and Kjetil A. KNARLAG[d]
[a]*Oslo Metropolitan University, Norway*
[b]*NAV, Norway*
[c]*University of Oslo, Norway*
[d]*Norwegian Directorate for Higher Education and Skills, Norway*
ORCiD ID: Frode Eika Sandnes http://orcid.org/0000-0001-7781-748X, Jo Herstad https://orcid.org/0009-0006-3807-1332

Abstract. This study explores the proposition of requiring students to hand in universally designed coursework and the transferrable benefits of accessibility audits. Coursework that adheres to universal design (UD) principles will be more accessible to fellow students and teachers. In this study we investigate if the universal design perspective can have positive side effects as a vessel for plagiarism detection. An experiment confirmed that an accessibility checking tool indeed can help flag some cheating attempts that go undetected by the plagiarism detection tool, but not all. Universal designed coursework requirements may prevent students from exploiting several of these cheating techniques. Through the process of preparing universally designed assignments, students will gain practice, knowledge, increased competence, and awareness of UD.

Keywords. Accessibility, universal design, education, coursework, plagiarism

1. Introduction

Accessibility legislation requires systems and contents to be universally designed. Public universities have a particular responsibility for ensuring universal design to support equal educational opportunities for all. Electronic learning materials and cloud-based systems including learning management systems and exam systems must therefore adhere to universal design principles [1, 2, 3]. Besides initiatives for establishing cultures for inclusion within education [4] and accessibility study programs [5], there are currently no norms, legislature, or common standards that regulate the accessibility of electronic coursework. Coursework can be classified as *personally generated content*. Coursework typically comprises electronic documents. Imagine if students had to satisfy the same strict universal design expectations as their teachers, teaching institutions, and future employers. Envisage that all student coursework submissions were automatically screened using accessibility-checking tools; coursework that adheres to the requirements would be approved for assessment while submissions with accessibility problems would need to be fixed and resubmitted before being assessed by teachers.

We wanted to explore this proposal in more depth as we were unable to identify coverage of this in the literature. Clearly, imposing such requirements on coursework will require students to acquire universal design competences. Skeptics could argue that this is an unrealistic proposal, yet universal design advocates may argue that this would train the students in universal design practices. Moreover, universally designed coursework would benefit the diverse cohorts of teachers and fellow students.

Pragmatists may argue that such proposals will ascertain little traction as most educators and education leaders have moderate or no interest in universal design. We therefore explore a potential by-product of universal design requirements; namely how universal design requirements may contribute to reducing plagiarism in written coursework. Automatic plagiarism detection has been discussed for several decades [6, 7], but has become more viable during the last decade due to cloud technologies. Several recent studies have evaluated plagiarism tools [8]. Elkhatat et al. [9] explored several attempts at bypassing nine academic plagiarism systems. Their attack vectors included text-as-images, invisible quotation marks, homoglyphs, and word concatenation with invisible symbols (letter *q* in white). Their results showed that none of the systems could detect all the attack vectors. Text-as-image and invisible symbols were the most challenging. StrikePlagiarism.com caught most attacks. The authors acknowledged that there is no empirical data about the extent of such cheating approaches. Alvi et al. [10] focused on methods for detecting homoglyph attacks, namely the use of homoglyph tables and a domain specific similarity measure. Plagiarism tools typically report some similarity score for each submission. Educators usually examine coursework above certain similarity limits for plagiarism. Manley [11] warned against this practice and recommended that educators should examine all the plagiarism reports.

The vast literature on cheating in higher education serves as a testament to the wide interest in this topic [6-12]. Jeergal et al. [12] reported that 76% of students admitted to having participated in cheating. Moreover, recent advances in artificial intelligence [13] and privacy are also much-discussed topics in context of cheating [14, 15]. This study emerged following the extensive media coverage of several cheating cases in Norway involving people in position of power.

This study explores how cheating attempts that pass undetected through a plagiarism tool can be detected by a tool intended for detecting accessibility violations [16]. Manipulative formatting is exploited that is visually unnoticeable to readers yet making the texts unrecognizable to the plagiarism detection tools. We argue that the benefits of universal design for cheating detection can help persuade skeptics to endorse universal design requirements on students' coursework.

2. Method

2.1. Ideation of test cases

We started with an ideation process inspired by ideas from computer security, namely phishing and search engine obfuscation where the users (or crawling robots) are misled through visual trickery. This resulted in the following main "attack vectors": disguising text-as-images, obfuscating texts using homoglyphs, and obfuscating text using invisible formatting including small spaces, color formatting and occlusion. The ideation was conducted independently of, and before exploring, the literature [9].

2.2. Educational context and pdf

Although most students prepare their reports in Microsoft Word there are still a small percentage that for various reasons (cost, ideology, training) use other word processors (Apple Pages, Google Docs, LibreOffice), or document preparation systems (Latex, Overleaf, InDesign). Therefore, pdf-files are commonly used as these can be generated by most tools and prevent teachers having to handle different, sometimes obscure, document formats.

2.3. Plagiarism detection

We used the Ouriginal plagiarism detection system integrated into the Canvas learning management system. Ouriginal is considered one of the leading plagiarism systems used by many higher education institutions in Norway. A national service provider (SIKT) coordinates the service and maintains the coursework database. When grading the submissions the educator will see a color-coded icon with a similarity percentage on the submission overview page. The educator also has the option of viewing a detailed plagiarism report for a given coursework.

2.4. Accessibility checking

We used the PDF/UA foundation PAC 2024 (version 24.2.1.0 beta) [17] for accessibility checks as it is a non-commercial tool endorsed by several disability organizations. Besides PDF/UA compliance, PAC 2024 also validates WCAG2.1 compliance. The WCAG2.1 reports are particularly useful as many readers probably are more familiar with WCAG2.1 than PDF/UA.

2.5. Procedure

For this study a dummy assignment was configured within a canvas course and the various test cases were submitted using the student test mode. After submission, it takes up to half an hour for the plagiarism report to be ready.

2.6. Material

As a test case an arbitrarily chosen paragraph of text comprising 183 words was taken from the introductory part of the PhD dissertation by Oddmund Hoel, the current Norwegian minister of education and research entitled “Målreising og modernisering i Noreg 1885–1940” (NTNU).

To validate that this text would trigger plagiarism warnings the passage was pasted into an empty Microsoft Word document (Office 365 version 2404, build 17531.20152) and saved as a pdf document. Word did not give any accessibility warnings during the pdf generation step. Acrobat reader (version 2024.002.20759) declared that document adhered to the PDF/A (archive) standard which may easily be confused with PDF/UA (accessibility). The plagiarism tool reported 100% similarity with the original document, confirming that the source document was included in the database.

The document was also tested with the accessibility tool to identify a baseline of issues common to all Word-generated documents (to be ignored herein). There were two

PDF/UA issues and one WCAG2.1 issue. The PDF/UA issues included “PDF/UA identifier missing” and “Title missing in document's XMP metadata”. The WCAG 2.4.2 violation was described as “Title missing in document's XMP metadata”. The title issues were eliminated by inserting a title into the document information dialog in Word. No simple way of resolving the “PDF/UA identifier missing” (within Word) was found.

3. Results

3.1. Case 1: Text-as-image

The document was printed, and the printout scanned using an office scanner. When feeding the scanned pdf into the plagiarism tool it showed a grey box with an explanation mark and a warning that the document did not contain any text. This should make the educator suspicious as the text can be seen, yet the tool states that there is no text.

When running the same document through the accessibility tool 9 WCAG issues were reported, namely 6 *perceivable* issues and 3 *operable* issues. These include 1.3.1 (Info and relationships, “document is not marked as tagged” and “tagged content and artefacts”), and 2.4.2 (“display of document title in the window title”). No issues related to missing alternative text were reported.

Next, instead of using a scanner, an image snapshot was taken of the text in the Word document and this image was pasted into a blank Word document and resized to visually look like the original. Next, the document was exported to pdf. During this export Word raised accessibility warnings related to the image alternative text. Word prompted us for an automatically generated alternative text to be checked and/or changed. The alternative text read: “A close-up of a document. Description automatically generated”.

When running this document through the plagiarism tool the same gray “no text” warning appeared. Next, the accessibility tool reported 3 WCAG issues related to criterion 1.3.1 ("Figure" element on a single page with no bounding box) and a warning related to criterion 4.1.1 (Possibly inappropriate use of a "Figure" structure element). The missing title was also highlighted. It may not be obvious for an assessor how these issues can be related to the text-as-image issue.

The automatically inserted alternative texts caused the document to pass the alternative text check on false premises. If removing this automatically generated alternative text, the accessibility checker indeed warns about the missing alternative text (WCAG 1.1.1, Alternative text missing for "Figure" structure element). Hence, the accessibility tool does not appear to be the problem, but rather the pdf-generation tool (Word) due to the automatically inserted alternative texts.

3.2. Case 2: Text-as-image with decoy text

To bypass the “no text detected” plagiarism tool warning the image of the text was accompanied by a brief 130-word passage of arbitrary text. This text was made invisible with white formatting [18]. This text passed with 0% similarity. However, if inspecting the detailed plagiarism report the decoy text is shown with the formatting stripped. Hence, a vigilant educator would catch the cheating attempt, while an educator that simply follows the overview indicator would not notice the attempt. However, the accessibility tool indeed detects the invisible text formatted with no contrast, namely WCAG criterion 1.4.3 - Text with insufficient contrast (distinguishable).

3.3. Case 3: Decoy text occluded by text-as-image

Since the color formatting triggered a contrast violation an attempt was made keeping the text color but instead hide the decoy behind the image of the plagiarized text. The pdf-document was saved as text using the Acrobat reader to confirm that the decoy text was included in the document. This attempt passed undetected by the plagiarism check. Moreover, the contrast violation was avoided. Only the missing bounding box violation gave a clue to the cheating attempt.

3.4. Case 4: Homoglyphs

Homoglyphs offer a well-known security attack mechanism [19]. Homoglyphs are characters that have the same visual appearance as others (i.e. the regular alphabet) but with a different digital representation. By replacing regular letters in the text with such homoglyphs the resulting text becomes unrecognizable. Unicode 0435, a lookalike for e, was used herein as e is the most frequent letter in both Norwegian and English. To prepare the document the search and replace function in Word was used to replace all the lowercase e characters with the homoglyph.

The plagiarism tool reported a 0% similarity. However, the detailed report provides a "mixed alphabet" warning suggesting that there could be an attempt to subvert the plagiarism analysis. The attempt passed unnoticed through the accessibility checker. According to WCAG criterion 3.1.1 (Language on Page) the language used should be specified with a correct ISO language code (e.g. "en-gb" for British English). To the best of our knowledge, most accessibility checking tools simply check for the presence of such language codes. One could argue that the addition of a test for illegal characters for a given language could help detect certain accessibility problems.

3.5. Case 5: Invisible dots

The text was altered by inserting dots between each character (as in "w.o.r.d.". Each dot was formatted to be invisible to readers by formatting the text to 1 point white. A simple script was written to perform this task. Hence, the text was pasted into the tool, and its result pasted back into Word. Word indeed warned about the lack of contrast when converting the document to pdf.

The plagiarism tool reported a similarity of 87% with other texts. Five different sources were listed but not the original document. An inspection of the matches seems to suggest that others have previously employed this approach as well. The detailed report also contained the text with the invisible formatting removed revealing the cheating attempt. Despite the high plagiarism score, the tool did not manage to identify the original source. The accessibility tool successfully identified the invisible text with 1240 instances of insufficient contrast (WCAG criterion 1.4.3).

3.6. Case 6: Invisible spaces

To avoid the suspicious looking dots in the plagiarism report and the detection of missing contrast by the accessibility tool a script was written that inserted a space at a random position in all words with more than 6 characters (long words). The inserted space changes the original word to two new non-words thereby sabotaging the pattern matching.

These spaces were made invisible with 1pt text size. The script is available at https://frode-sandnes.github.io/textObfuscator/. Hence, the text was first copied into the form and the result pasted back into Word. Word did not report any accessibility issues.

The plagiarism tool reported 0% plagiarism with no warnings. However, if inspecting the detailed report, one may notice the odd spaces at various positions in the text. The accessibility tool did not report any issues besides the missing title. Hence, size formatting passes undetected although they represent a noteworthy accessibility problem.

4. Discussion

4.1. Can we rely on plagiarism and accessibility tools?

Results confirm that it is relatively easy to bypass plagiarism checks with simple means. Splitting words with invisible spaces was probably the approach least likely to draw attention from educators. The plagiarism tool seemed to perform approximate text matching well but with less attention to the overall context of use.

The accessibility checks did uncover some of the irregularities introduced by the visual trickery, but not all. In sum, the tool was successful in identifying text disguised using color formatting (insufficient contrast). The accessibility checker would thus also likely detect the invisible character attacks reported by Elkhatat et al. [9]. However, the tool was unsuccessful in detecting text disguised using tiny text formatting. Such texts are probably illegible to most readers, not just readers with reduced visual acuity. On the other hand, this result appears correct as WCAG does not give absolute font size recommendations (besides for color contrast limits). The gist is that a (web) document should be resizable allowing readers to choose the text size. Clearly, a pdf document is generally zoomable. For instance, the Acrobat reader allows a document to be magnified by 6400%. Note that only space characters will appear invisible if color formatting is not used, as all other symbols (such as alphabetic letters) will be visible to the readers after closer inspection even if these are given a small font size.

The plagiarism tool successfully warned about the homoglyphs, while these passed undetected by the accessibility tool. It seems the accessibility tool did not check for language specifications, or lack thereof. Leaning on WCAG 3.1.1 one could argue that such tools should flag the use of symbols from scripts not matching the current language.

The tests revealed that text-as-images passed undetected by both the plagiarism detection tool and the accessibility checking tool. The latter was due to the authoring tool (Word) inserting placeholder alt texts that resulted in a pass for the alternative text checks. This could suggest a larger problem as Word is commonly used for generating pdf-documents, in that it will generate pdf-documents that pass automatic WCAG alternative text checks, while not containing alternative texts that provide any useful information for screen readers. Functionality that was intended to help authors create accessible documents indeed becomes an accessibility problem.

However, the inserted images resulted in other accessibility issues. A trained evaluator may learn to read such signs. On the other hand, a student report with many illustrations may overwhelm the evaluator with violations making it hard to separate accessibility issues from cheating attempts.

The accessibility tool was unable to detect the occlusion issues where the decoy text was hidden behind an image. The lack of occlusion accessibility checks has been pointed out by others [20], and several approaches for detecting occlusions in the context of

rendering failure detection have been proposed [21]. In sum, it seems we cannot fully rely on either plagiarism detection tools or accessibility tools. Combined, the tools can uncover more cheating attempts.

4.2. Limitations of this study

This study is based on the technology available in 2024. Plagiarism and accessibility technologies are continuously being developed and improved and it is likely that one will get different results if repeating this study as the technologies evolve. Next, a handful of attack vectors were explored. This is not an exhaustive list of cheating approaches.

Although the visual cheating attempts explored herein are relatively straightforward to execute, they also require some technical insight. Hence, the explored scenarios are hypothetical as we, and others [9], do not know the extent to which such techniques are exploited in practice. It would be relevant to uncover if such techniques are utilized, and if so, to what degree. Such insights could be obtained by analyzing past coursework.

4.3. Recommendations and future directions

Besides using accessibility tools, how can educators detect such cheating attempts? And, if detected, how can one identify the plagiarized sources? A practical approach is to open a pdf document in Acrobat reader and save the result as text. The resulting file will reveal visual tricks such as invisible characters since the text is stripped of formatting. Moreover, this text file will also include all the alternative texts in the document. Automatically generated placeholders can then be detected through manual inspections.

To locate the plagiarized source the plagiarism tool needs to be fed the text as perceived by the readers. Elkhatat et al. [9] suggest that this can be achieved by rendering the documents as images and then extracting the text using optical character recognition (OCR). Theoretically, OCR software will not respond to invisible characters, will treat homoglyphs as the most typical characters with a particular appearance, and extract text from images. A simple test was conducted to validate this claim by feeding each test case into a Tesseract OCR demo (https://tools.simonwillison.net/ocr). The results confirmed that each extracted text gave close to 100% similarity with the source.

If students could only submit coursework that adheres to accessibility guidelines, students attempting to use certain visual cheating attempts would be stopped due to the accessibility violations. Hence, in addition to promoting universal design, the requirement may have a preventive effect on cheating.

Future work includes shedding light on the underlying reasons and rationale for why students were exempt from the UD requirements. Moreover, it would be relevant to probe different (technical and non-technical) student cohorts' perceptions towards coursework UD requirements, as well as how this would align with student organizations' work on promoting equal access to education for all.

5. Conclusions

This study explored the simplicity with which a plagiarism tool can be bypassed using visual trickery, and to what degree an accessibility tool can help detect such cheating attempts. The results show that the accessibility tool could identify attempts at making text invisible through color manipulations but was unable to detect text size

manipulations. Also, it failed to detect text-as-images, homoglyphs, and visually obstructed text but gave indirect clues to manipulations manifested in bounding box violations which require experience to interpret successfully. We argue for requiring students to provide universally designed documents to raise their awareness of accessibility and to prevent cheating.

References

[1] Burgstahler SE. Universal Design in Higher Education: From Principles to Practice. Harvard Education Press; 2015.

[2] Fornauf BS, Erickson JD. Toward an inclusive pedagogy through universal design for learning in higher education: A review of the literature. J Postsecond Educ Disabil. 2020;33(2):183-199.

[3] Saplacan D, Herstad J, Pajalic Z. Feedback from digital systems used in higher education: An inquiry into triggered emotions. In: Transforming Our World through Design, Diversity and Education: Proceedings of Universal Design and Higher Education in Transformation Congress; 2018 Oct. (Vol. 256).

[4] Olaussen EJ, Heelan A, Knarlag KA. Universal Design for Learning–license to learn: A process for mapping a Universal Design for Learning process on to campus learning. In: Transforming higher education through Universal Design for Learning. Routledge; 2019. p. 11-32.

[5] Whitney G, Keith S, Bühler C, Hewer S, Lhotska L, Miesenberger K, et al. Twenty five years of training and education in ICT Design for All and Assistive Technology. Technol Disabil. 2011;23(3):163-170.

[6] Jian HL, Sandnes FE, Huang YP, Cai L, Law KM. On students' strategy-preferences for managing difficult course work. IEEE Trans Educ. 2008;51(2):157-165.

[7] Foltýnek T, Meuschke N, Gipp B. Academic plagiarism detection: a systematic literature review. ACM Comput Surveys (CSUR). 2019;52(6):1-42.

[8] Foltýnek T, Dlabolová D, Anohina-Naumeca A, Razı S, Kravjar J, Kamzola L, et al. Testing of support tools for plagiarism detection. Int J Educ Technol High Educ. 2020;17:1-31.

[9] Elkhatat AM, Elsaid K, Almeer S. Some students plagiarism tricks, and tips for effective check. Int J Educ Integrity. 2021;17:1-12.

[10] Alvi F, Stevenson M, Clough P. Plagiarism detection in texts obfuscated with homoglyphs. In: Advances in Information Retrieval: 39th European Conference on IR Research, ECIR 2017, Aberdeen, UK, April 8-13, 2017, Proceedings 39. Springer International Publishing; 2017. p. 669-675.

[11] Manley S. The use of text-matching software's similarity scores. Account Res. 2023;30(4):219-245.

[12] Jeergal PA, Surekha R, Sharma P, Anila K, Jeergal VA, Rani T. Prevalence, perception and attitude of dental students towards academic dishonesty and ways to overcome cheating behaviors. J. adv. clin. res. insights. 2015); 2(1): 2-6.

[13] Elkhatat AM, Elsaid K, Almeer S. Evaluating the efficacy of AI content detection tools in differentiating between human and AI-generated text. Int J Educ Integrity. 2023;19(1):17.

[14] Brinkman B. An analysis of student privacy rights in the use of plagiarism detection systems. Sci Eng Ethics. 2013;19:1255-1266.

[15] Sandnes FE. On portfolio assessment, group work, and quasi-anonymization: What structural information do anonymized reports reveal? In: Proceedings of ICITL 2024, LNCS. Springer (In press); 2024.

[16] Tollefsen M, Ausland T. A practitioner's approach to using WCAG evaluation tools. In: 2017 6th International Conference on Information and Communication Technology and Accessibility (ICTA). IEEE; 2017 Dec. p. 1-5.

[17] PAC 2024 - PDF Accessibility Checker. [Internet]. 2024 [cited date]. Available from: https://pac.pdf-accessibility.org/en/download/

[18] Sandnes FE. Understanding WCAG2. 0 color contrast requirements through 3D color space visualization. Stud Health Technol Inform. 2016;229:366-375.

[19] Sern LJ, David YGP, Hao CJ. PhishGAN: Data augmentation and identification of homoglyph attacks. In: 2020 International Conference on Communications, Computing, Cybersecurity, and Informatics (CCCI). IEEE; 2020 Nov. p. 1-6.

[20] Sandnes FE. "Consent notices are obstructing my view": Viewing sticky elements on responsive websites under the magnifying glass. Displays. 2024;81:102579.

[21] Liu Z, Chen C, Wang J, Huang Y, Hu J, Wang Q. Nighthawk: Fully automated localizing ui display issues via visual understanding. IEEE Trans Softw Eng. 2022;49(1):403-418.