Comparing and Combining Tests for Plagiarism Detection in Online Exams

Edward F. Gehringer, Xiaohan Liu, Abhirav Dilip Kariya, and Guoyi Wang
North Carolina State University
{efg, xliu74, akariya, gwang25}@ncsu.edu

ABSTRACT

Online exams with machine-readable answers open new possibilities for plagiarism and plagiarism detection. Each student's responses can be compared with all others to look for suspicious similarities. Past work has developed several approaches to detecting cheating: n-gram similarity, Levenshtein distance, Smith-Waterman distance, and binomial probability. To that we add our own term-frequency based approach, called the "weirdness vector," which measures how unusual a student's answers are, compared to all other students. Each of these approaches seems suited to particular question types. Levenshtein and Smith-Waterman are suited to long text strings, as appear in answers to essay questions. Binomial probability and n-gram similarity are well suited for finding suspicious patterns in responses to multiple-choice questions. The "weirdness vector" is most applicable to fill-in-the-blank questions.

Unlike past research, that applied a single metric to detect cheating in an exam with questions of a single type, this paper measures how different approaches work with different kinds of questions, and proposes methodologies for combining the approaches for exams that consist of all three kinds of questions. This work shows promise for detecting cheating in open-web exams, where students can cheat using covert Internet channels, and is especially applicable in situations where exams cannot be proctored.

Keywords

Online exams; plagiarism; Levenshtein distance; n-grams

1. INTRODUCTION

Online exams have become more common in recent years due to the growth in online courses, especially after the transition to emergency online instruction. They have the advantage of faster grading, especially for distance ed, more copious feedback, and they can provide a more authentic testing environment by allowing students to access certain information from the web (e.g., the course notes) during the exam.

Yet open-web exams do raise concerns about cheating [1]. Browsers can be locked down, and students can be monitored remotely with cameras [2]. But monitoring is expensive, and locking down browsers may destroy the authenticity of the environment. For example, in a course on open-source coding, students would always do their work online. If they don't have access to the Internet during an exam, they must work in an environment far different from their usual one. However, an authentic testing environment can only be used if there is a way to detect plagiarism.

Our approach is to use data mining to measure the similarity of the submitted answers. We extend our past work [3] by incorporating additional published tests into our application, and studying their applicability to different types of questions. Section 2 covers tests that have been proposed by others. Section 3 introduces new techniques for handling particular kinds of questions. Section 4 reports our findings from experiments on real data, and discusses which metrics are suitable for which types of questions. Section 5 summarizes our work and points out ideas for future progress.

2. RELATED WORK

Many published papers address automated detection of plagiarism, but with few exceptions, each paper focuses on a single mathematical test. While a few papers [4] do consider multiple tests, they do so in the context of comparing competing tests for detecting plagiarism on a particular kind of question (e.g., multiple choice). Since exams contain many different kinds of questions (multiple choice, essay, fill in the blank, matching, etc.) what is needed is a single application that can apply appropriate tests to responses to different kinds of questions. That is the goal of our research.

2.1 Levenshtein Distance

The Levenshtein distance between two strings is the minimum number of edits required to change one string into the other. For example, the Levenshtein distance between "faculty" and "faulty" is 1, the Levenshtein distance between "sloop" and "sleep" is 2, and the Levenshtein distance between "country" and "countries" is 3. In the research of investigating whether a machine learning model based on a statistical method works better than a model based on a structural method, the Levenshtein distance was chosen to be the similarity measurement for the structural approach.

Edward Gehringer, Xiaohan Liu, Abhirav Kariya and Guoyi Wang "Comparing and combining tests for plagiarism detection in online exams" In: *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*, Anna N. Rafferty, Jacob Whitehill, Violetta Cavalli-Sforza, and Cristobal Romero (eds.) 2020, pp. 605 - 609

Levenshtein distance has been researched not only for traditional string match, but as a structural method in clustering-based machine learning models of plagiarism detection [5].

One limitation of Levenshtein distance in detecting plagiarism is that rearrangement of text produces a large Levenshtein distance, since Levenshtein distance is focused on one-character (or one-word) edits. Suppose that two students' answers, taken as a whole, bear little resemblance to each other, but they contain sequences in different positions that are highly similar. The Smith-Waterman algorithm can identify this.

2.2 Smith-Waterman Algorithm

The Smith-Waterman algorithm is another classical string similarity metric. It looks for similar local regions to identify optimal sequence alignments. For example, the best alignment of two sequences X = "abcbadb" and Y = "abbdb" would be

a b c b a d b
a b - b - d b

Researchers have proposed alterations of the Smith-Waterman algorithm that were tested effective in practice of detecting collusion while speeding up the algorithm without using up much space [6]. Traditional Smith-Waterman algorithm searches through a pair of sequences and finds the maximum piece of consecutive matching characters, whereas the revised implementation introduces the cut-off concept to keep track of multiple pieces of matching. The modification yields more optimal local alignments and thus more effective on plagiarism detection as well.

2.3 n-grams

Another attempt from the structural perspective is n-grams. We can consider a word as a token [7]. Then an n-gram is a set of n consecutive words. Then for two exam submissions, we can ask what is the longest common n-gram between them, or how many n-grams of length $> k$ do they have in common? This is a useful metric for comparing two students' essay answers, but it also useful for comparing other kinds of answers, such as answers to multiple-choice (MC) questions. Here, MC answers, not words, make up the strings we are comparing.
MC questions have the property that the answers are chosen from a discrete set, usually about four in cardinality. Given that there are m possible answers for each question, the probability that two students will choose the same answer by chance is $\frac{1}{m}$. The probability that they will choose the same k consecutive answers is $\frac{1}{m^k}$. This is the idea behind the binomial test [8]; it is very unlikely that two students will choose a large number of the same wrong answers by chance.

Each of these methods works well on a specific type of text. A more comprehensive approach that works on all types of questions is needed for online exams. We will further analyze the effectiveness of each metric to determine which metrics work better for multiple-choice questions, fill-in-the-blank questions, and essay questions, respectively.

3. PROPOSED METHODS

3.1 The "weirdness" vector

The weirdness-vector metric looks for pairs of students who have similar but unusual answers. The basic idea is to calculate the term frequency of each response by each student and create a vector of term frequencies. Then we can use cosine similarity to measure the distance between the weirdness vectors of each pair of students. Those who have the most similar vectors are worth further inspecting.

3.1.1 Data Preprocessing

1. For the set of students $S = s_1, s_2, \ldots, s_n$, we extract all their responses R into a matrix where $r_{i,j}$ is the response to question q_i by student s_j .

2. Then we remove the stop words and punctuation in the response matrix.

3. We use a function to classify each question on the exam as belonging to one of three question types: Multiple-choice, fill-in-the-blank, and free-response essay questions.

3.1.2 Implementation

1. For each response $r_{i,j}$ of student s_j to question q_i, we calculate its term frequency among all the responses to question q_i . Each response $r_{i,j}$ is converted into a "bag of words," and is compared with every other bag-of-words response to question q_i. The number of occurrences of each bag of words divided by the number of students n gives us the frequency $f_{i,j}$ of a response $r_{i,j}$.

$$f_{i,j} = \frac{\text{number of times } r_{i,j} \text{ appears in responses to } q_i}{n}$$

2. It is the low term frequencies that may be suspicious, but for the other tests in the program, high values are suspicious. Hence, we calculate the inverse term frequency instead:

$$w_{i,j} = 1 - f_{i,j}$$

3. Each student s_j has a "weirdness" vector W_j consisting of the inverse frequencies $w_{i,j}$ of each response to each question q_i , i.e., $W_j = w_{1,j}, w_{2,j}, \ldots, w_{m,j}$, where $q_1, q_2, \ldots, q_m$ are the questions in the test.

4. We use cosine similarity to measure the closeness between pairs of weirdness vectors. For a pair of vector X and Y, the cosine similarity is calculated as

$$\text{cosine similarity} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$$

where x_i and y_i , $i = 1, 2, \ldots, n$ are components of X and Y. The vectors with similarly high inverse frequencies will return high cosine similarity, for small values in the vector components do not contribute much when calculating the cosine similarity. That being said, only similar but unusual responses will stand out in similarity scores.

3.1.3 *Regarding identical answers*

The weirdness vector highlights suspicious behavior by detecting pairs of exams that contain identical incorrect answers, yet it is also worthwhile to identify pairs of exams that have identical correct answers. Some questions have multiple possible answers, where students can answer correctly without necessarily providing the exact same responses. Such cases can be well addressed by an algorithm that takes multiple correct answers into account; however, no algorithm can detect plagiarism among students who have provided the same correct response where only one correct response is possible.

3.2 Bag-of-Words Extension

Several metrics help detect plagiarism in text-based answers. Results can be enhanced by preprocessing the text before applying metrics. One kind of preprocessing is getting rid of stop words and removing punctuation. We can go one step further and treat the remaining words as an unordered set. This is the bag-of-words model.

3.2.1 *Use Case*

To illustrate the advantage of the bag-of-words model for finding similar answers, consider this example:

Response_1 = "pattern: strategy"

Response_2 = "strategy pattern"

Response_1 == Response_2 // False

bag_of_words(remove_stop_words(Response_1)) ==

bag_of_words(remove_stop_words(Response_2))// True

Given that these responses are deemed incorrect, it is worthwhile to count the two wrong answers as matching. Without the removal of stop words and bag-of-words analysis, this case would go unnoticed as evidence of potential plagiarism.

4. EMPIRICAL RESULTS

The research questions that we are trying to answer are whether the tests can detect suspicious similarity, as well as which tests are most effective on each type of questions. We consider a test effective if it produces only a few unusual values (outliers) among its results. Of course, results from tests alone cannot be solid evidence of cheating; instructors would need to inspect the exam papers. To forestall excessive manual inspection, a good test should direct attention to the few most suspicious responses. If the observed values given by a metric contain outliers when it is applied to a particular kind of question, this metric can be deemed useful for that type of question.

We can use data visualizations to illustrate the effectiveness of all the metrics on three types of questions. We used real exam data from CSC 517 (all offerings between Fall 2014 and Spring 2020) at North Carolina State University. All data was de-identified before use.

4.1 Effectiveness of each metric

The weirdness metric shows us (Figure 1) how unusual it is for a pair of exams to share the same wrong answer to a question. Weirdness is a good test for FiB exams if only a few exams have a large number of the same unusual incorrect

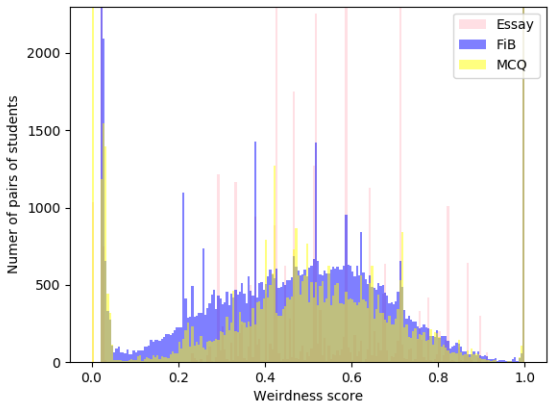

Figure 1: Weirdness metric on 3 types of questions

answers. On essay questions, however, it is the responses with high term frequency that are suspicious, since each response should have its unique phrasing. Essentially, each incorrect essay response is considered "weird" and hence, the weirdness values will show a discrete distribution as shown in the histogram above. For MC questions, there is a much smaller number of possible choices, and thus, weirdness does not work as effectively as for FiB. Though both the FiB and MC weirdness values have small tails, the values are more meaningful for FiB questions.

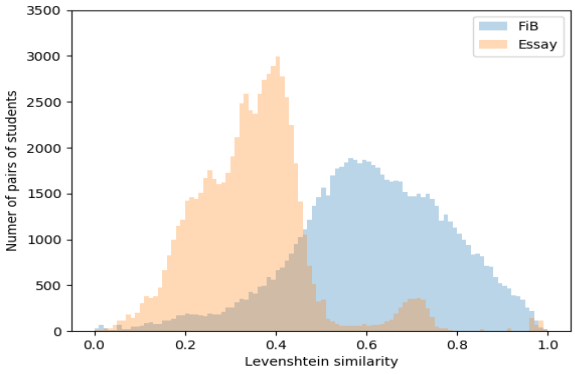

Figure 2: Levenshtein on FiB and essay questions

The Levenshtein metric (Figure 2) uses edit distance to compute string similarity. While weirdness watches for short unusual responses for FiB questions, Levenshtein has its strength in detecting long similar responses for essay questions. As the histogram for Levenshtein performance shows, Levenshtein generates many fewer outliers on essay questions than on FiB questions. The essay histogram has a minor peak near 1.0, highlighting the responses that are suspiciously similar, whereas the FiB histogram has many high values, suggestive of false positives.

Smith-Waterman is pretty good at comparing long texts, and it is much more revealing on essay questions than on

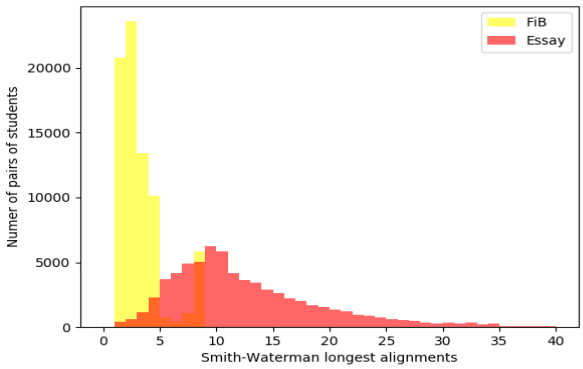

Figure 3: Smith-Waterman metric results

FiB questions, as we can tell from the graph above.

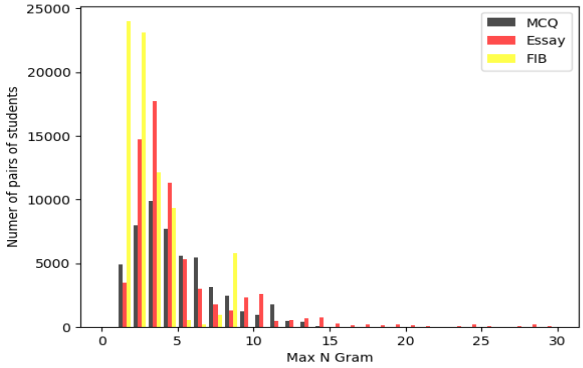

Figure 4: Max n-grams metric results

n-grams are naturally suited to finding long similarities in essay questions. We can also concatenate all the MC responses of two students and use n-grams to compute the longest common subsequence between them. Since the number of pairs drops significantly after max n-gram length = 10 for essay questions, we choose 10 as the threshold and consider those greater than 10 to be outliers. FiB responses are much shorter, typically no longer than 7 words, and are expected to be mostly identical; thus n-grams are unlikely to provide much guidance. The max n-grams lengths of MC responses tell us how many consecutive MC questions two students answered identically.

The n-gram metric can, of course, help detect students who were collaborating extensively on MC questions, but it does not take correctness of the responses into account. Consecutive same correct MC responses should not be treated as suspicious.

Binomial is used for MC questions only, as it calculates the probability of students having the same wrong answers. It is a more reasonable metric for MC questions than n-grams because it does take correctness of responses into account.

4.2 The most suitable metric

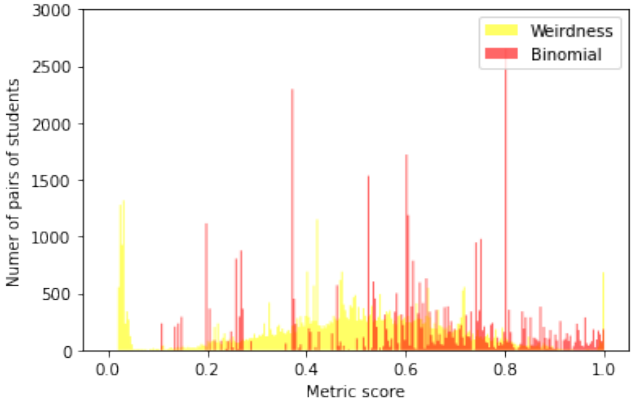

Figure 5: Different metrics on MC questions

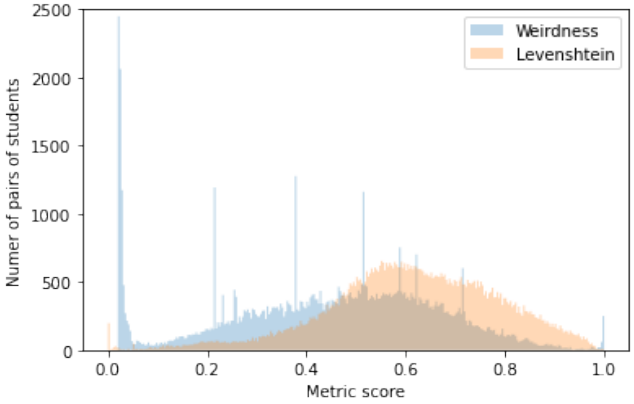

Figure 6: Different metrics on FiB questions

As discussed earlier, weirdness is much more applicable to FiB questions than other string matching metrics. The concave upward curve at (0.8, 1.0) justifies the effectiveness of weirdness.

Levenshtein, Smith-Waterman, and N-grams are all good metrics for essay questions, although empirically, Levenshtein is more effective over other tests.

5. SUMMARY

We can conclude from the empirical results that for multiple-choice questions, one should seek help from the binomial test. For fill-in-the-blank questions, weirdness works the best. For essay questions, Levenshtein, Smith-Waterman, and n-grams all work effectively.

6. REFERENCES

[1] G. Fenu, M. Marras, and L. Boratto, "A multi-biometric system for continuous student authentication in e-learning platforms," *Pattern Recognition Letters*, vol. 113, pp. 83–92, 2018.

[2] Y. Atoum, L. Chen, A. X. Liu, S. D. Hsu, and X. Liu, "Automated online exam proctoring," *IEEE Transactions on Multimedia*, vol. 19, no. 7, pp. 1609–1624, 2017.

[3] S. Biswas, D. G. Edward F. Gehringer, S. Sahane, , and S. Sharma, "A data-mining approach to detecting plagiarism in online exams," in *EDM 2018, Proceedings of the 11th International Conference on Educational Data Mining*, pp. 504–507.

[4] C. Zopluoglu, "Similarity, answer copying, and aberrance: Understanding the status quo," *Handbook of quantitative methods for detecting cheating on tests*, pp. 25–46, 2017.

[5] E. Anzén, "The viability of machine learning models based on levenstein distance and cosine similarity for plagiarism detection in digital exams," 2018.

[6] R. W. Irving, "Plagiarism and collusion detection using the smith-waterman algorithm," *University of Glasgow*, vol. 9, 2004.

[7] M. Zini, M. Fabbri, M. Moneglia, and A. Panunzi, "Plagiarism detection through multilevel text comparison," in *2006 Second International Conference on Automated Production of Cross Media Content for Multi-Channel Distribution (AXMEDIS'06)*, pp. 181–185, IEEE, 2006.

[8] F. S. Belleza and S. F. Belleza, "Detection of cheating on multiple-choice tests by using error-similarity analysis," *Teaching of Psychology*, vol. 16, no. 3, pp. 151–155, 1989.