Improving mathematics assessment readability
Do large language models help?

Patel, Nirmal; Nagpal, Pooja; Shah, Tirth; Sharma, Aditya; Malvi, Shrey; Lomas, Derek

DOI
10.1111/jcal.12776

Publication date
2023

Document Version
Final published version

Published in
Journal of Computer Assisted Learning

Received: 3 December 2021 | Revised: 8 July 2022 | Accepted: 22 December 2022

DOI: 10.1111/jcal.12776

ARTICLE

Journal of Computer Assisted Learning WILEY

Improving mathematics assessment readability: Do large language models help?

Nirmal Patel[1] | Pooja Nagpal[2] | Tirth Shah[1] | Aditya Sharma[1] | Shrey Malvi[1] | Derek Lomas[3]

[1]Playpower Labs, Gujarat, India

[2]Central Square Foundation, Delhi, India

[3]Industrial Design Engineering, Delft University of Technology, Delft, Netherlands

Correspondence
Nirmal Patel, Playpower Labs, Gujarat, India.
Email: nirmal@playpowerlabs.com

Abstract

Background: Readability metrics provide us with an objective and efficient way to assess the quality of educational texts. We can use the readability measures for finding assessment items that are difficult to read for a given grade level. Hard-to-read math word problems can put some students at a disadvantage if they are behind in their literacy learning. Despite their math abilities, these students can perform poorly on difficult-to-read word problems because of their poor reading skills. Less readable math tests can create equity issues for students who are relatively new to the language of assessment. Less readable test items can also affect the assessment's construct validity by partially measuring reading comprehension.

Objectives: This study shows how large language models help us improve the readability of math assessment items.

Methods: We analysed 250 test items from grades 3 to 5 of EngageNY, an open-source curriculum. We used the GPT-3 AI system to simplify the text of these math word problems. We used text prompts and the few-shot learning method for the simplification task.

Results and Conclusions: On average, GPT-3 AI produced output passages that showed improvements in readability metrics, but the outputs had a large amount of noise and were often unrelated to the input. We used thresholds over text similarity metrics and changes in readability measures to filter out the noise. We found meaningful simplifications that can be given to item authors as suggestions for improvement.

Takeaways: GPT-3 AI is capable of simplifying hard-to-read math word problems. The model generates noisy simplifications using text prompts or few-shot learning methods. The noise can be filtered using text similarity and readability measures. The meaningful simplifications AI produces are sound but not ready to be used as a direct replacement for the original items. To improve test quality, simplifications can be suggested to item authors at the time of digital question authoring.

KEYWORDS

GPT-3, mathematics assessment, readability, text simplification

1 | INTRODUCTION

Readability is the ease with which a reader can understand a written text. Many factors contribute to the readability of a text, including personal interests, content, style, and text organization (Dale & Chall, 1949). In the digital world, the concept of readability also extends to how text is displayed on screens. Readability is essential when it comes to mathematics assessments. If test developers produce math test items at inappropriate reading levels, the questions can become biassed (Walkington et al., 2018). If the test takers find the test item challenging to read, they will likely get it wrong. Readability issues can also affect item quality metrics, such as difficulty and discrimination. If a math test item is too difficult to read, it can end up measuring reading comprehension in addition to the math skill. Therefore, readability is one of the many factors that influence the validity of an assessment.

Research on mathematics assessments has indicated that word problems are notorious for their difficulty (Cummins et al., 1988). Teachers often cite word problems as a significant weakness of students (Loveless et al., 2008). A subset of word problems is story problems in "real world" contexts referencing concrete people, places, and objects. If these contexts are unknown to the test-takers, they can get confused about what the question is asking. Whilst researchers have found that story problems often make mathematics easier (Koedinger & Nathan, 2004), this is only possible when the readability of the items is appropriate for the test takers.

Researchers have analysed the reading difficulties of assessment items and how they relate to the test outcomes. Lamb (2010) analysed State assessments from Texas to discover that reading levels of the test items were overlapping across grades. Dempster and Reddy (2007) examined the readability factors of TIMSS (Trends in International Mathematics and Science Study) science assessments administered in English. They analysed three factors - sentence complexity, unfamiliar words, and long words to investigate the readability issues in the test. The researchers found that sentence complexity negatively influenced learners' performance on TIMSS items, resulting in random guessing and favouring an incorrect option. The negative effect was more pronounced in learners with limited proficiency in English than those who were more proficient in English. Walkington et al. (2018, 2019) showed that English Language Learners (ELL) find it more challenging to answer math word problems than non-ELL students.

Lamb (2010) showed a correlation between readability and math achievement. They argued that readability issues in summative math tests could put some students at a severe disadvantage. Lamb also provided a way to correct the test scores based on the readability analysis. Nandhini and Balasundaram (2012) designed a framework to classify math word problems based on readability to help learners with reading difficulties. They presented a naive-Bayes classification model that predicted readability and showed that their model was better at predicting the assigned grade level of the text. Predicting the assigned grade level of the text may not be entirely useful since the text can already be misaligned with the grade level in its original classification. King and Burge (2015) did a readability analysis of PISA 2012 assessments and found that the PISA 2012 assessment was at the appropriate reading level. Interestingly, they combined all of the words in the math test and analysed them as a complete unit since individual problems had less than 30 words. This is different from other works where individual items were analysed separately, regardless of the number of words in them.

The demands of solving math word problems can be explained by cognitive load theory. Cognitive load refers to the amount of mental effort expended on a task (Sweller et al., 1998). Applied to the context of word problems, Intrinsic cognitive load refers to the inherent difficulty of understanding particular mathematical concepts in a word problem. Extraneous cognitive load refers to how the material is presented, or the learning-irrelevant activities required of students (Sweller et al., 1998, p. 259). Texts that are difficult to read may increase extraneous cognitive load as students struggle to decode the written language to form a situation model.

1.1 | Automated text simplification

We wanted to understand whether large language models help us simplify difficult-to-read math word problems. Large language models have demonstrated impressive capabilities in generating natural language. These models are growing in size, with some having more parameters than the number of neurons in the human brain. One of the largest and most commonly known language models, GPT-3, has 175 billion parameters (compared to an average of 85 billion neurons in the human brain). Typically, text data from the internet is used to train large language models. Once the parameters are learned, the model can be adapted to many different tasks. The adaptation capability also has left many wondering whether this demonstrates a general intelligence system. But as many experts know, the language models are far from reliable. Dale (2021) called GPT-3 an "unreliable narrator" who has no obligation to tell the truth. This is quite right, as the text that the model produces is generated based on randomized procedures.

Large language models are usually designed as sequence to sequence models. They take a text input sequence up to a certain length, and the model tries to complete the input sequence. We can leverage the general prompt completion capability of the GPT-3 engine to perform different functions, including text simplification. Consider the example below in Table 1.

Whilst the output starts with some relevant sentences, it gets derailed in another direction. We can either tune the model to make the output better or only consider the relevant parts of the output. We decided to use the GPT-3 model to produce easier to read versions of existing math word problems.

If we can generate simplifications of math word problems through the AI engine, they can be used to improve existing assessment content and augment question authoring interfaces. Automated Text Simplification is an active area of NLP research. A recent review paper on Text Simplification (Janfada & Minaei-Bidgoli, 2020) noted three

TABLE 1 Text simplification using GPT-3. The maximum output length was 100 tokens

An example of text simplification by GPT-3 (italics = input, following text = output)
My second grader asked me what this passage means: """ *Jupiter is the fifth planet from the Sun and the largest in the Solar System. It is a gas giant with a mass one-thousandth that of the Sun, but two-and-a-half times that of all the other planets in the Solar System combined. Jupiter is one of the brightest objects visible to the naked eye in the night sky, and has been known to ancient civilizations since before recorded history. It is named after the Roman god Jupiter.[19] When viewed from Earth, Jupiter can be bright enough for its reflected light to cast visible shadows,[20] and is on average the third-brightest natural object in the night sky after the Moon and Venus*. """ *I rephrased it for him, in plain language a second grader can understand*: """ Jupiter is the fifth planet from the Sun and the biggest planet in the Solar System. It's a big ball of gas. It's very bright and it can be seen from Earth at night. Jupiter is named after the Roman god Jupiter. I think this is a good example of why we should teach science to children in plain language. I'm not sure how many of us learned science by reading a textbook, but if we did, we probably did not understand it.

distinct types of simplifications that AI systems can do: lexical, syntactic, and semantic. The lexical changes are related to changing words and replacing them with simpler words. The syntactic changes rearrange the components of the sentences. And the semantic changes can change the meaning of the text. We can leverage this generalized simplification capability of the text models to improve the readability of the math word problems.

Text simplification systems have many similarities with the language translation systems. In both of these systems, the main task is to produce a translation of the input. The simplification system can consider the simple to read output written using another type of "simplified language". Several open-source datasets are available to measure the capabilities of the machine simplification systems, including a Wikipedia dataset that aligns article text from the regular Wikipedia to the Simple Wikipedia (Hwang et al., 2015). The text simplification training datasets consist of input-output pairs where inputs are difficult to read and outputs are easier to read. GPT-3, on the other hand, can work without any training data or with little training data. We took advantage of this capability of the model and used it to simplify math word problems. Given that the underlying simplification approach is similar to translation, we consider the words simplification, translation, rephrasing, and paraphrasing to mean the same thing: simplifying the text and improving its readability. If we can simplify math word problems and ensure that they are at the appropriate grade reading level, the math assessments will provide us with more reliable measures of student learning. To improve the readability of word problems, we can measure the readability levels of the input and simplified texts and see if there is a significant improvement.

1.2 | Readability measurement

There are many measures of readability. Prins and Ulijn (1998) define readability as "the ability of the text to communicate the writer's intention to the intended reader". Rakow and Gee (1987) define readability as "an estimate of the probability of comprehension by a particular group". Several different classes of methods are available to assess the readability of a text. François and Fairon (2012) summarized the following classes of the readability measures:

1. Formula-based approaches – use text properties as part of a regression formula.
2. Structuro-cognitive approaches – consider non-superficial features of the text such as coherence, content density, the flow of the ideas (Corlatescu et al., 2021), and so forth.
3. AI readability – prediction models that take text as input and predict a numeric readability score as output; these models use machine learning or deep learning methods.

Additionally, there is a fourth class of instruments for assessing text comprehensibility in the form of questionnaires (Friedrich & Heise, 2019; Sadoski et al., 1993; Sadoski et al., 2000).

To date, formula-based readability measures are most widely used in practical applications. One of the main reasons for their popularity is their relative simplicity and availability of free software implementations. In their essential components, all traditional methods for computing readability are similar. They tend to incorporate some combination of easily measured units like sentence length, word length, and word frequency (Benjamin, 2012). Several tools are available to compute readability measures used in Structro-cognitive approaches. Common Text Analysis Platform (CTAP) (Chen & Meurers, 2016), Coh-Metrix (Graesser et al., 2004), TACCO (for measuring Cohesion) (Crossley et al., 2016), and TAALES (for measuring Lexical Sophistication) (Kyle & Crossley, 2015) are some of the tools freely available online. Many measures from structro-cognitive approaches have been used in AI readability models as predictors.

When we try to improve the readability of math word problems, we can measure the improvement by computing the difference between the readability scores of the output and input. We can use more than one metric to ensure that we are not getting false positives. However, there is one challenge with using only readability measures. If the resulting simplification is simple but unrelated to input, it is not useful. To make sure that the simplifications of AI are related to the input, we can use text similarity measures.

The rest of the paper is organized as follows: the Related Work section outlines studies around the readability analysis of mathematics texts and manual and automated ways of improving the readability of educational texts. The Study Design section describes how we leveraged the text simplification capabilities of the GPT-3 engine. The Data section describes how we sampled the math word problems from the open-source EngageNY curriculum and how we used the GPT-3 API (an Application Programming Interface to access GPT-3 capabilities via HTTP protocol) to simplify all of the sampled problems.

The Metrics section gives details of the Readability and Text Similarity metrics that are used to find useful simplifications. The Method section is brief and describes how we used Cohen's d to measure improvements in readability metrics. The Results section describes how GPT-3 performed in the simplification task. The Discussion section gives some ideas on how the AI engine may be functioning internally to make the simplifications and how it can be used in a real-world scenario. The Future Work section describes the next set of work that needs to be done to make our method more accurate so that it can deliver real-world impact.

2 | AUTOMATED READABILITY IMPROVEMENT OF MATH TEXTS

Researchers have identified many issues with mathematics text readability. These issues go beyond syntax and cover vocabulary, diagrams, symbols, rhetorical questions, and information organization (Noonan, 1990). Several studies have analysed the readability of math texts.

If the test items have poor readability, we can revise them to improve their reading difficulty. There are many different methods to measure text readability, but some methods may be more valid than others when revising instructional texts. Davidson and Kantor (1982) found that making readability improvements based on readability formulas could do more harm than good. They discovered that "adaptations were found to be most successful when the adaptor functioned as a conscientious writer rather than someone trying to make a text fit a level of readability defined by a formula". Britton and Gülgöz (1991) used a structuro-cognitive model of text comprehension to manually improve text readability. They compared the improved version of the text with the original on undergraduates and found the improved version had higher recall rates. This showed that authors could systematically improve their text by using appropriate readability measures.

With recent advances in natural language processing methods, researchers have attempted to automatically improve text readability. One of the works on automated text simplification was done by Chandrasekar and Srinivas (1997), who split the input sentences into shorter sentences to improve readability. This improvement approach is more aligned with readability formulas because they heavily rely on average sentence length for readability measurement. So shorter sentences would produce higher readability scores. We previously described the lexical, syntactic, and semantic simplifications in the Text Simplification section. Recent survey papers on text simplification have outlined fundamental approaches and studies in this area (Al-Thanyyan & Azmi, 2021; Janfada & Minaei-Bidgoli, 2020). In the context of the educational texts, De Belder and Moens (2010) presented lexical and syntactic approaches to simplify texts which will help students who are struggling to read understand the text better. Nandhini and Balasundaram (2013) proposed simplifying text by extracting relevant and easy-to-read sentences from the original text. Recently, Rebello et al. (2019) surveyed students from grades 2 to 4 and provided a statistical analysis about how syntactic simplification of text comprehension improved the reading ability and performance of the students.

Our work primarily focused on using the GPT-3 system to simplify math word problems. As per our literature review in this field, no other studies have used the GPT-3 AI to simplify educational texts. To understand how a large language model can simplify math word problems, we designed a study that used multiple ways to simplify input texts.

3 | STUDY DESIGN

We wanted to find out whether the GPT-3 AI can improve the readability of the math word problems. We did several manual studies within the GPT-3 engine to observe its behaviour. We saw that the AI sometimes gave great simplifications, but it did not always give outputs that carried all of the key information from the input. This led us to focus on finding meaningful simplifications that could be used in a real-world context.

Although we do not have direct control over how the GPT-3 simplifies the given text, we can use readability and text similarity measures that tell us whether the given simplification is meaningful or not. A meaningful simplification would show improvements in readability whilst still having at least semantic similarity with the input. It is possible that the syntax of the input might have to be changed considerably to improve the readability. If the metrics suggest that the simplification is not useful, we can sample another text from the engine and continue to do so until we find a useful simplification.

We designed three types of prompts to simplify the math word problems so that we can observe the difference in their behaviour and understand which prompts led to more desirable results. These prompts contained the math word problem and the instructions for the AI engine to produce the simplification. Table 2 presented below outlines these prompts.

We can find the first two prompts (Rephrase and Rephrase with Title) on the OpenAI documentation (OpenAI, n.d.-a, n.d.-b). These prompts are presented as summarization prompts rather than simplification prompts. Text summarization is usually aimed at converting a large amount of text into a much smaller amount by omitting the little details. This is contrary to what we need to do when trying to make a math word problem easy to read. We need to keep all minute details intact whilst ensuring that the text is readable. The first two prompts are "open-ended" and do not give any explicit instructions to the AI engine regarding how the simplifications should be made. To provide instructions to the GPT-3, we can use the few-shot learning prompts.

Few-shot learning refers to learning from very few examples. GPT-3 became known because of its capability of few-shot learning. This is indeed the holy grail of AI, to learn a task from a very few examples. We wanted to understand how well few-shot learning worked for the word problem simplification task. The third prompt was designed by one of the researchers, where they created five training examples for the AI. These examples were simplifications of five randomly selected problems from the dataset. These training examples are listed in Appendix A.

TABLE 2 Examples of the three types of prompts we used in the study. Prompts are input texts for the GPT-3 model. The outputs are not shown

Prompt	Prompt text (italics = input word problem, output is not presented)
Rephrase	*Serena had 30 candies and Mill had 15 candies. How many pieces of candies did Serena and Mill have in total?* I rephrase this for my daughter, in plain language a second grader can understand:
Rephrase with Title	My second grader asked me what this passage means: """ *Serena had 30 candies and Mill had 15 candies. How many pieces of candies did Serena and Mill have in total?* """ I rephrased it for him, in plain language a second grader can understand: """
Few Shot Learning	Simplify the given text. ### Text: Luis uses square inch tiles to build a rectangle with a perimeter of 24 inches. Does knowing this help him find the number of rectangles he can build with an area of 24 square inches? Why or why not? Simplified: You want to count all the ways to make 24 square inch rectangles. You first make a 24 inch perimeter rectangle with 1 square inch tiles. Will this help? ### Text: Fill in the missing whole numbers in the boxes below the number line. Rename the whole numbers as fractions in the boxes above the number line. Simplified: Write whole numbers in the boxes below the number line. Write fractions equal to whole numbers in the boxes above the number line. ### Text: Compare the perimeter of your tessellation to a partner's. Whose tessellation has a greater perimeter? How do you know? Simplified: Compare your pattern with someone else's pattern. Which pattern has a longer perimeter? ### Text: *Serena had 30 candies and Mill had 15 candies. How many pieces of candies did Serena and Mill have in total?* Simplified:

We created further variations of all three prompts. Table 3 shows variations of the prompts in Table 2. For Rephrase and Rephrase with Title prompts, we had a choice to tell the AI which reading grade level we were aiming for. The GPT-3 system is able to identify the grades when mentioned in the input, and it translates the input text accordingly. For example, you can "ask" the system to rephrase the same passage to a second and an eighth-grader, and you will get outputs with different reading levels. To leverage this "understanding" of grade levels that GPT-3 has, we decided to construct prompts that tried improving the reading level by one and two grade levels. The input math word problems were collected from grades 3 to 5, and we created multiple simplification prompts from them accordingly. For example, if the math word problem was from the 3rd grade, the AI tried to read it to second and first graders.

3.1 | Research questions and hypotheses

We wanted to validate whether the GPT-3 AI was able to perform the meaningful simplification of math word problems. For the simplification to happen, the readability of the text passage must improve. Not only that, the simplified text passage should contain all important information and context provided in the input passage – that is, the simplification should be meaningful. We hypothesize that GPT-3 will be able to do meaningful simplification of the word problems. We divided our overall hypothesis into two hypotheses:

Hypothesis 1. *The GPT-3 AI will improve the raw readability scores of the math word problems.*

Hypothesis 2. *The GPT-3 AI will provide meaningful simplifications of math word problems.*

We also had additional research questions concerned with automatically identifying meaningful simplifications:

RQ1. Can we identify meaningful simplifications using numeric metrics?

RQ2. Are some numeric metrics particularly useful in indicating meaningful simplifications?

The H1 was only concerned with finding whether some or all prompts could improve readability as measured by our metrics. H2 was more focused on finding the meaningful simplifications that could be used in an actual test. We collected math word problems data and simplified them using GPT-3 to test our hypotheses.

4 | DATA

We collected input word problems from an open-source curriculum, and we simplified these word problems using the GPT-3 API.

4.1 | Math word problems

We collected the math word problems from the EngageNY open-source math curriculum. This curriculum is available for everyone under the Creative Commons licence. We focused on word problems from grades 3 to

TABLE 3 Conditions of the study, each condition was a prompt template that contained the problem to be rephrased and the instructions for the AI on how to do it. The stop sequences are the character sequences that tell the GPT-3 AI to stop producing outputs

Condition code	Prompt	Variation	Stop sequence
1	**Rephrase** One line direction after input.	Improve readability by one grade level	
2		Improve readability by two grade levels	
3		To second grade level	
4	**Rephrase with Title** Two line directions, input sliced between directions.	Improve readability by one grade level	“””
5		Improve readability by two grade levels	“””
6		To second grade level	“””
7	**Few Shot Learning** A few examples for AI to see.	5 examples	###

5, as the authors had prior experience working with students and data from those grades. We downloaded all of the mathematics module PDFs from the EngageNY website and created a process to extract the word problems. The paper also refers to the word problems as passages or excerpts because we assumed that all multi-line passages extracted from assessment pages were either parts of the questions or the entire question. Our step-by-step process is outlined in Table 4.

Since we were analysing an already published curriculum where only the PDF files were available, we had to resort to a long scanning and cleaning process that was not perfect. Ideally, we can avoid this process if the passages to be examined are available as digital text. We found that the OCR process did not work well with math symbols. This led to several improperly detected word problems that were left out in step 8. Systems like GPT-3 are not trained to understand complex math symbols. Hence we did not improve the OCR detection rate further. We limited our analysis to problems with words, integers, and decimals.

TABLE 4 The step-by-step process for collecting math word problems from EngageNY grades 3 to 5 math curriculum

Step	Description	Notes
1	Downloaded all module PDFs for Math grades 3 to 5.	From www.engageny.org
2	Converted PDF file pages into PNG files using ImageMagik.	Image resolution was kept 1275 px × 1650 px.
3	Text was extracted from PNG files using PyOCR.	OCR was not fully accurate.
4	Only kept pages having “Homework”, “Exit Ticket”, “Problem Set”, or “Assessment” in their titles.	
5	Extracted text passages by joining lines without spaces between them.	
6	Removing garbage passages that do not have any educational content. Left with a dataset of 1994 passages.	These are headers, footers, standard or skill codes, and so forth.
8	Selected properly formatted passages using a set of regex filters (rules and filter creation methodology described in Appendix B). Left with a dataset of 1384 clean passages.	This filter was ~80% accurate (training accuracy) to detect properly formatted passages. The accuracy was calculated by first hand selecting 262 grade 5 passages and then using regex to predict the hand selections.
9	Randomly sampled $N = 250$ passages from the universe of 1384 clean passages.	Only these passages are used for analysis.

4.2 | GPT-3 simplifications

Once the $N = 250$ passages to be analysed were collected, we created a design matrix containing seven GPT-3 prompts for every input passage. Each of these prompts corresponded to one condition in Table 3. Once we constructed the design matrix, we iterated over each prompt and used the GPT-3 API to receive the simplifications. Table 5 shows our GPT-3 API parameters:

We had seven different prompts for every input word problem that attempted to simplify it. And for every prompt, we requested three different simplifications for each prompt. Some simplifications began with a stop sequence (as denoted in Table 5), and we discarded them. We did not attempt to collect more non-null simplifications for the input passage. This led to some input passages resulting in less than three simplifications. Once we collected the outputs by calling the API, they were saved for further analysis.

GPT-3 AI translated each of the $N = 250$ input word problems into easier-to-read versions using all seven prompts. This led to a total possible output of 5250 excerpts. We discarded the engine's blank and failed responses, which led to $M = 4992$ output passages. The approximate cost for this data collection was $35. The Table 6 shows how many outputs each condition had:

TABLE 5 GPT-3 API parameters that we used

API parameter	Description	Value
Engine	Type of the engine. OpenAI has several types of engines to make inferences from GPT-3. We used the latest engine available at the time of our analysis.	davinci-instruct-beta-v3
n	Number of simplifications or prompt completions to attempt.	3
Temperature	Variation in output, low temperature will result in similar outputs.	0.8
Prompt	Input prompt that GPT-3 will try to complete. These prompts were created for each input sentence.	Precomputed based on the condition.
Max_tokens	How many maximum tokens GPT-3 can output. According to their documentation, 75 words roughly corresponds to 100 words. We did not want more words in the output than the input.	Number of words in the input × 1.3333
Stop	The stop sequence, GPT-3 will stop producing further tokens if the stop sequence is generated.	As shown in Table 3
Frequency_penalty	Penalty parameter to stop same words from appearing again and again in the output.	0.2

Before analysing the data collected from GPT-3, we passed it through preprocessing steps below:

1. Replaced all of the newline characters with spaces
2. Removed extra spaces from the beginning and end
3. Removed last incomplete sentences from all passages
4. Removed the blank responses

We decided to remove the last incomplete sentence because we would expect the AI only to provide complete suggestions in a real-world context. We did not have any grammatical correction programme that we could use to correct the sentences, so we removed the last sentence regardless of its content. We can attempt to "repair" the sentence using grammar correction software to keep the maximum amount of AI-produced information.

5 | METRICS

In this section, we describe the Readability and Text Simplification metrics used in our analysis. We need to use Readability metrics to measure the improvement in readability and Text Simplification metrics to ensure that the simplification is meaningfully related to the input.

5.1 | Readability metrics

Bertram and Newman (1981) discussed the weaknesses of classical readability measures. Their first concern was that most readability formulas deal only with sentence length and word difficulty—they tend to ignore other factors like cohesion or the complexity of ideas. Sawyer (1991) summarized many similar viewpoints, describing the shortcomings of the readability formulas for revising instructional texts. The highly simplistic nature of readability formulas may not capture all of the complexities of educational texts. This concern was addressed by the structure-cognitive measures of readability (François & Fairon, 2012). These measures were understandable metrics designed to understand the linguistic properties of the texts, such as Lexical Diversity and Cohesion (or lexical linking). Bertram and Newman's second concern is the lack of consideration for reader-specific factors like reader interest or their purpose for reading. This concern is truly

TABLE 6 Number of inputs and outputs for each condition

Inputs	Condition	# of simplifications	Number of outputs (out of 750 total possible)
250	Rephrase One Grade Level Below	3	733
250	Rephrase Two Grade Levels Below	3	735
250	Rephrase for Second Grader	3	741
250	Rephrase With Title One Grade Level Below	3	738
250	Rephrase With Title Two Grade Levels Below	3	731
250	Rephrase With Title for Second Grader	3	737
250	Few Shot Learning 5 Examples	3	586

Note: Total unique inputs: $N = 250$; Total GPT-3 API calls = 250 × 7 × 3 = 5250; Total successful outputs = 4992; Total blank responses and API failures = 258.

worth addressing and may lead to improved student outcomes if addressed.

In addition to these concerns, there are other limitations to readability measures that prevent their use in practice to improve the readability of assessment word problems. For instance, to generate meaningful results, most readability measures require a minimum of 300 words, yet many word problems involve fewer than 300 words. In our readability analysis of input problems, we found that Cohesion metrics did not have any meaningful variance in them to compare the readability scores of the passages. Further, many readability measures use standard lists of familiar words. This approach to determining familiarity does not consider the fact that some words in an assessment may have been explained as part of the text, either in text or through the use of a glossary. Additionally, readability analysis does not consider the text's design explicitly. Texts used in assessments may include drawings, graphs, tables, and so forth. However, it is likely that these additional elements are referred to in the text. Given these limitations, we need to pick the metrics and interpret the results carefully.

Calculating the readability metrics requires using a tool or a programme that implements the algorithms to calculate the measures. We used the Common Text Analysis Platform (or CTAP) tool to compute various readability metrics that measured the linguistic properties of the text. The goal for AI was to either increase or decrease these measures based on whether the higher value meant higher or lower readability. The CTAP tool provides a user interface to upload a large text corpus and calculate several features for those texts. The tool offers close to 1000 metrics. Another publicly available tool, Coh-Metrix 3.0, provides several hundred metrics for text readability.[1] We attempted to find common metrics between both frameworks, and used CTAP to derive the metrics. Coh-Metrix does not provide a web interface to generate metrics for a large corpus.

We chose sixteen metrics to analyse the readability of the input and output passages (described in Table 7). These metrics have been shown to be significant predictors of human-rated readability scores, grade levels of the text, or the reading ability of the learners. The first set of metrics we used was Descriptive, measuring things such as the number of tokens and average sentence length. These metrics have been used in many readability formulas. For example, the Flesch Reading Ease formula (Flesch, 1948) has average words per sentence and average syllables per word as predictors. Nelson et al. (2012) noted that all of the Descriptive measures that we used (# 1–5 in Table 7) were negatively correlated with readability, that is, the metric would go down with an increase in readability. Next, we used three Lexical Diversity metrics (# 6–9 in Table 7), one of which (# 7) was found to be stable for shorter texts (Zenker & Kyle, 2021). All three metrics we used have been assessed for their validity by Kyle et al. (2021), and researchers found them to be negatively correlated with human judgements of lexical diversity. Next, we used four Part of Speech features that counted the number of nouns, verbs, adjectives, and adverbs (# 9–12 in Table 7). These features were hypothesized to be negatively correlated with reading grade level measures by Heilman et al. (2008). Pitler and Nenkova (2008) found number of verbs per sentence (which is directly proportional to the number of verbs) to be significantly correlated with text readability ratings. The rest of the four features (# 13–16 in Table 7) were related to Lexical Sophistication, which measures things like how easy it is to imagine the words in the text and how easy it is to point to real-world objects denoted by the given words. All of these features have been found to correlate with the Lexical Proficiency of the learners (Kyle & Crossley, 2015). Although we see that all of the metrics have been found to be negatively correlated with readability judgements of humans, experiments have shown that the characteristics of readers can also affect the readability of the text. McNamara and Kintsch (1996) found that expert readers learned more from abstract texts that had some challenging elements, whilst novice readers learned and remembered more from easy-to-read texts. This means that the Lexical Sophistication measures that we are using may have more application in the context of novice readers rather than experts.

We defined readability improvement as follows:

$$\text{Readability Improvement} = \text{Readability Score}\,(\text{OutputText}) - \text{Readability Score}\,(\text{InputText})$$

Based on the results of the prior studies, we expected all of our readability metrics to decrease for the output texts.

5.2 | Text similarity metrics

We used two different metrics to compute the similarity of input word problems and their simplifications. The first metric computed the percentage of common words between the input and the output. The second metric calculated cosine similarity between the text embeddings of the input and output. The embeddings were produced by GPT-3 API. Both of these metrics are described below.

5.2.1 | Percentage of common words

This metric is relatively simple and is defined as follows:

$$\%\ \text{Common Words} = \frac{\text{Num Common Words}\,(\text{Input, Output})}{\text{Num Total Words}\,(\text{Input, Output})} \times 100$$

Both the common words and total words were counts of unique words. If two different words had the same root word (e.g., river and rivers), they were counted as different words. This metric can be considered valid for discovering useful math word problem simplifications because we would expect the AI to retain the key information in the input. It is possible that AI changes the words but keeps meaning the

[1] https://cohmetrix.memphis.edu/cohmetrixhome/documentation_indices.html

TABLE 7 Readability metrics from CTAP tool used in our analysis

Category	No.	Readability metric	Description	Expected improvement direction
Descriptive	1	Number of Tokens	Calculates the number of tokens in the text.	Negative
	2	Mean Sentence Length in Tokens	Calculates the mean sentence length in number of tokens.	Negative
	3	Mean Token Length in Syllables	Calculates the mean token length in syllables.	Negative
	4	Mean Token Length in Letters	Calculates the mean token length in letters.	Negative
	5	Number of Sentences	Calculates the number of sentences in a text.	Negative
Lexical Diversity	6	Lexical Richness: Type Token Ratio (Root TTR Words)	Calculates the Type Token Ratio (TTR) for words (excluding punctuation and numbers) of a text. TTR is the ratio of the total number of unique words divided by the total number of tokens in the passage.	Negative
	7	Lexical Richness: Type Token Ratio (Root TTR)	Type Token Ratio that considers all tokens (including punctuation and numbers).	Negative
	8	Lexical Richness: MTLD (excluding punctuation and numbers)	MTLD stands for Measure of Textual Lexical Diversity and calculates lexical diversity of the text for all words excluding punctuation and numbers.	
Word Information	9	Number of POS Feature: Noun Tokens	Calculates the number of noun tokens in the text. POS refers to Part of Speech.	Negative*
	10	Number of POS Feature: Verb Tokens (including Modals)	Calculates the number of verb tokens including modals in the text.	Negative
	11	Number of POS Feature: Adjective Tokens	Calculates the number of adjective tokens in the text.	Negative*
	12	Number of POS Feature: Adverb Tokens	Calculates the number of adverb tokens in the text.	Negative*
	13	Lexical Sophistication Feature: Age of Acquisition	Calculates the approximate Age of Acquisition by using a norm word data from the Medical Research Council (MRC) Psycholinguistic Database.	Negative
	14	Lexical Sophistication Feature: Meaningfulness	Calculates the Meaningfulness of the words by using norm data form MRC Psycholinguistic Database. A word is more meaningful when it is more related to other words.	Positive
	15	Lexical Sophistication Feature: Imageability	Calculates Imageability based on the norm word data from the MRC Psycholinguistic Database. A word has higher imageability if it is easy to imagine it for example, apple. Abstract words are more difficult to imagine.	Positive
	16	Lexical Sophistication Feature: Concreteness	Calculates Imageability based on the norm word data from the MRC Psycholinguistic Database. A word is more concrete if it can be pointed to easily.	Positive

*Improvement hypothesized in the literature but significant change not reported.

same. In those cases, the next metric which measures semantic similarity may be more helpful.

5.2.2 | Cosine similarity of text Embeddings

Text embeddings are becoming increasingly common in NLP applications. The key idea behind text embeddings is the vectorization of text onto a relatively small number of latent dimensions (compared to possible words in the lexicon). The word embeddings convert a word into a vector. Different entries of this vector can represent different latent concepts. We can combine word embeddings in many different ways. You can take the mean of all of the word embeddings in a sentence, and you can get a sentence embedding. Another way to get sentence embedding is to supply embeddings of every word as an input into a recurrent neural network. The word vectors are passed through a neural network, and they are combined through the information propagation mechanism of the model. In the end, we get one numeric vector for the input text – the text embedding vector. We can do the same for a bigger block of text (i.e., small or big passages).

Text embedding vectors can help us model the latent dimensions contained in the input text. If two pieces of text are the same, their embedding vectors will be identical. Similarly, different sentences will have different embedding vectors. We can use cosine similarity measures to calculate how close two sentences are. Nandhini and Balasundaram (2012) used cosine similarity to filter out unimportant details from the input sentence. This was a novel way to simplify the text. In our study, we used cosine similarity to measure the closeness of the input and output sentences. We expected that a reasonable simplification would be where AI would change the arrangement of the words without modifying the underlying meaning too much. If you produce the same text as the simplification of the input, it would be entirely similar to the input, but this is not what we want when we are doing the simplification. We would expect some difference in the text if there is a considerable improvement in the readability. We used GPT-3's Embeddings API to generate the vector representations of the world problems. Then, we measured the similarity between input and output text as follows:

$$\text{Embedding Similarity} = \text{Cosine Similarity}(\text{Embedding}(\text{Input Text}), \text{Embedding}(\text{Output Text}))$$

6 | STATISTICAL ANALYSIS

We measured the readability improvement in the text passages using Cohen's d. For every input passage, multiple simplifications were generated. We calculated the standardized difference in the means for every readability metric mentioned in the Metrics section. We first calculated the readability metrics of the input passages. Then we calculated the readability metrics of the output passages. There were multiple outputs for the same input passage, and this led to repeated metric values for the input passage. For example, if an input passage had a Lexical Richness value of 15 units, it is possible that its translations had Lexical Richness values of 12, 15, or 18 units. In this case, we considered input passages appearing three times in the sample and each output passage occurring one time. This meant that the sample sizes for the input and output passages were the same, and input passages had repeated metrics. We calculated Cohen's d using the psych package in the R environment.

To find meaningful improvements in text passages, we relied on text similarity metrics and improvement metrics calculated as differences of the output readability metric minus the input readability metric. We applied threshold values on the calculated metrics (similarity and readability improvement) to find meaningful improvements that had face validity.

We used a simple threshold-based approach to find meaningful simplifications. The difference in readability metrics (output minus input) can be used to identify simplifications with desirable properties. We applied multiple combinations of filtering conditions to find meaningful simplifications. Each set of filters had the following conditions in common:

- Input and output should end the same way (question mark or full stop)
- % of common words > = 70
- Cosine similarity of >0.85 and <1.0

7 | RESULTS

Let us begin by looking at some interesting simplifications of word problems that GPT-3 generated (Table 8). We discovered these by applying threshold values to readability improvement and similarity metrics. These examples are available in our open-source data (provided in Section 10).

We can see in the table that the algorithm can improve the input in different ways. The first and third examples have simplified sentences, whilst the second example breaks down one sentence in the input into two sentences in the output. The fourth and fifth examples have many verbs and adverts added in the simplification.

7.1 | Readability improvement

For each simplification, we generated several metrics that defined the overall quality of the simplification. To understand the overall behaviour of the GPT-3 engine, we can look at the standardized difference in the mean values of the readability metrics. The table below shows Cohen's d for each of the metrics, indicating the difference in the standardized mean of the readability metric between the input and the output. We calculated the difference both across conditions and separately for each condition.

There are several ways to interpret the effect size. Our table contains both positive and negative effect sizes. Sawilowsky (2009) suggested a new rule of thumb, suggesting that d(0.5) = medium,

TABLE 8 Selected examples of GPT-3 simplifications

#	Input	Simplified by GPT-3
1	Vivienne draws four identical rectangles as shown below to make a new, larger rectangle. The perimeter of one of the small rectangles is 18 cm, and the width is 6 cm. What is the perimeter of the new, larger rectangle?	Vivienne draws four identical rectangles to make a new, larger rectangle. Perimeter of one small rectangle is 18 cm. Width of one small rectangle is 6 cm. What is the perimeter of the new, larger rectangle?
2	Draw a tape diagram to represent the following problem. Use numbers to solve and write your answer as a statement. Check your answer.	Draw a tape diagram to represent the problem. Use numbers to solve the problem. Write your answer as a statement. Check your answer.
3	It takes Greg 15 min to mow the front lawn. It takes him 17 more minutes to mow the back lawn than the front lawn. What is the total amount of time Greg spends mowing the lawns?	It takes Greg 15 min to mow the front lawn. It takes him 17 more minutes to mow the back lawn. What is the total amount of time?
4	Use the information in the chart below to plan an environment for your robot. Write the width and length for each rectangular item. Use the blank rows if you want to add extra circular or rectangular items to your robot's environment.	This passage is a chart with different shapes and sizes listed. It is asking you to plan an environment for your robot using the information in the chart. The width and length for each rectangular item can be used to create a space for your robot.
5	An airplane is descending into an airport. When its altitude is 5 miles, it is 275 miles from the airport. When its altitude is 4 miles, it is 200 miles from the airport. At 3 miles, it is 125 miles from the airport.	An airplane is flying closer and closer to an airport. When it is 5 miles away from the airport, it is 275 miles away from the airport. When it is 4 miles away from the airport, it is 200 miles away from the airport.

d(0.8) = large, d(1.2) = very large, and d(2.0) = huge. For metrics # 1–13, we had expected a negative effect. We can see in Table 9, except for the Mean Sentence Length in Tokens, readability metrics had a negative effect meaning that AI improved them in the expected direction. Figure 1 shows a visualization of effect sizes and their confidence intervals. For the last three metrics in the table (# 14–16), we had expected a positive effect. Here we saw that the Meaningfulness feature had a negative effect, whilst the other two had a small positive effect.

Figure 2 below shows Improvement *v/s* Input for all of the metrics (all conditions for each metric are combined in the visualization). In the visualization, we can see that if the input metric indicates more complexity, the AI shows more improvements on average.

Table 9 shows that for most of the readability metrics, the average improvement in readability was in the expected direction except for the two metrics: Mean Sentence Length in Tokens (#2) and Meaningfulness (#14). The effect sizes of each metric are different, the largest ones being the Number of Sentences and Number of Nouns. Figure 2 shows that the Number of Sentences did not have variance like other variables, so its effect size may not be stable. We can also see that the two variables with an average improvement in unexpected directions have differences amongst conditions. The Few Short Learning condition showed improvements completely consistent with prior literature. Figure 2 also shows that the Lexical Sophistication features had a different trend in the data, where values close to the median input readability saw several outliers. These outliers were present in all conditions. We also observed that if the readability of the input was low, there was "more room for improvement". For metrics except for Lexical Sophistication, when the input value was higher (indicating a difficult-to-read passage), then the improvement was more. The results in Table 9 and Figure 2 clearly indicate that GPT-3 AI was able to make improvements in the expected direction across all metrics. The data confirm H1 and show that The GPT-3 AI will improve the raw readability scores of the math word problems.

7.2 | Meaningful readability improvement

It is very easy to get meaningless improvements in GPT-3. Improvement in readability metrics does not concretely tell us whether the improvement was meaningful or not. For example, consider the following simplification that has reduced the number of sentences from four to one:

- *Input*: Fiona draws a 24 square centimetre rectangle. Gregory draws a 24 square inch rectangle. Whose rectangle is larger in area? How do you know?
- *Output*: Fiona has drawn a rectangle with a total area of 24 square centimetres.

We can clearly see that the reduction in sentence count cannot tell us much about the quality of the output. The output here is clearly not useful. We observed that, in general, one or two metrics were insufficient to indicate a successful simplification that could be considered for rewriting. In fact, any amount of improvement in readability cannot tell us whether the simplification is useful or not. We need to know how similar the simplification is to the input. Similarity can tell us whether parts of input were preserved in the output or not. When rephrasing math problems, we can combine the readability and text similarity metrics to find simplifications that are both more readable and similar to the input.

TABLE 9 Cohen's d for measuring improvement in readability metrics

		Cohen's d (intervals are 95% confidence intervals)							
No.	Readability metric	All Conditions	C1	C2	C3	C4	C5	C6	C7
1	Number of tokens	−0.98	−1.11 (−1.22, −1)	−1.06 (−1.17, −0.95)	−1.14 (−1.25, −1.03)	−0.91 (−1.01, −0.8)	−0.94 (−1.05, −0.83)	−0.98 (−1.08, −0.87)	−0.71 (−0.82, −0.59)
2	Mean sentence length in tokens	0.35	0.37 (0.27, 0.48)	0.36 (0.25, 0.46)	0.34 (0.24, 0.44)	0.48 (0.37, 0.58)	0.38 (0.28, 0.48)	0.43 (0.33, 0.54)	−0.09 (−0.21, 0.02)
3	Mean token length in syllables	−0.47	−0.52 (−0.63, −0.42)	−0.59 (−0.69, −0.48)	−0.57 (−0.68, −0.47)	−0.45 (−0.56, −0.35)	−0.43 (−0.53, −0.32)	−0.47 (−0.58, −0.37)	−0.21 (−0.32, −0.1)
4	Mean token length in letters	−0.43	−0.49 (−0.6, −0.39)	−0.54 (−0.65, −0.44)	−0.5 (−0.6, −0.4)	−0.47 (−0.57, −0.36)	−0.43 (−0.53, −0.32)	−0.46 (−0.56, −0.36)	−0.1 (−0.22, 0.01)
5	Number of sentences	−1.16	−1.31 (−1.43, −1.2)	−1.25 (−1.36, −1.14)	−1.32 (−1.43, −1.21)	−1.23 (−1.34, −1.12)	−1.17 (−1.28, −1.06)	−1.23 (−1.34, −1.12)	−0.58 (−0.69, −0.46)
6	Lexical richness: Type token ratio (Root TTR words)	−0.83	−1.02 (−1.13, −0.91)	−0.98 (−1.09, −0.88)	−1.01 (−1.12, −0.9)	−0.7 (−0.81, −0.6)	−0.72 (−0.83, −0.62)	−0.73 (−0.83, −0.62)	−0.66 (−0.78, −0.55)
7	Lexical richness: MTLD (excluding punctuation and numbers)	−0.46	−0.52 (−0.62, −0.41)	−0.59 (−0.7, −0.48)	−0.59 (−0.69, −0.48)	−0.34 (−0.45, −0.24)	−0.4 (−0.5, −0.29)	−0.37 (−0.47, −0.27)	−0.42 (−0.54, −0.3)
8	Lexical richness: Type token ratio (Root TTR)	−0.9	−1.1 (−1.21, −0.99)	−1.06 (−1.17, −0.95)	−1.08 (−1.19, −0.97)	−0.78 (−0.88, −0.67)	−0.79 (−0.9, −0.69)	−0.79 (−0.9, −0.69)	−0.67 (−0.79, −0.55)
9	Number of POS feature: Noun tokens	−1.05	−1.16 (−1.27, −1.05)	−1.14 (−1.25, −1.03)	−1.23 (−1.34, −1.11)	−1.02 (−1.13, −0.91)	−1.03 (−1.14, −0.92)	−1.1 (−1.21, −0.99)	−0.63 (−0.75, −0.51)
10	Number of POS feature: Verb tokens (including modals)	−0.77	−0.87 (−0.97, −0.76)	−0.84 (−0.94, −0.73)	−0.91 (−1.02, −0.8)	−0.71 (−0.81, −0.6)	−0.72 (−0.82, −0.61)	−0.73 (−0.83, −0.62)	−0.6 (−0.72, −0.49)
11	Number of POS feature: Adjective tokens	−0.57	−0.59 (−0.7, −0.48)	−0.56 (−0.66, −0.46)	−0.6 (−0.7, −0.49)	−0.55 (−0.66, −0.45)	−0.62 (−0.73, −0.52)	−0.62 (−0.73, −0.52)	−0.4 (−0.52, −0.28)
12	Number of POS feature: Adverb tokens	−0.4	−0.44 (−0.54, −0.33)	−0.46 (−0.57, −0.36)	−0.42 (−0.53, −0.32)	−0.32 (−0.42, −0.22)	−0.35 (−0.45, −0.24)	−0.36 (−0.46, −0.25)	−0.48 (−0.6, −0.36)
13	Lexical sophistication feature: Age of acquisition	−0.24	−0.3 (−0.41, −0.18)	−0.31 (−0.43, −0.2)	−0.36 (−0.48, −0.25)	−0.24 (−0.36, −0.13)	−0.2 (−0.31, −0.09)	−0.21 (−0.32, −0.1)	−0.06 (−0.19, 0.06)
14	Lexical Sophistication Feature: Meaningfulness	−0.1	−0.12 (−0.22, −0.02)	−0.15 (−0.26, −0.05)	−0.05 (−0.15, 0.05)	−0.13 (−0.23, −0.03)	−0.15 (−0.26, −0.05)	−0.12 (−0.22, −0.01)	0.05 (−0.06, 0.17)
15	Lexical sophistication feature: Imageability	0.06	0.01 (−0.09, 0.11)	0.01 (−0.09, 0.11)	0.14 (0.03, 0.24)	0.06 (−0.04, 0.16)	0.01 (−0.09, 0.12)	0.03 (−0.07, 0.13)	0.21 (0.09, 0.32)
16	Lexical sophistication feature: Concreteness	0.07	0.01 (−0.09, 0.11)	0.02 (−0.09, 0.12)	0.12 (0.01, 0.22)	0.08 (−0.03, 0.18)	0.03 (−0.07, 0.13)	0.05 (−0.05, 0.16)	0.2 (0.08, 0.31)

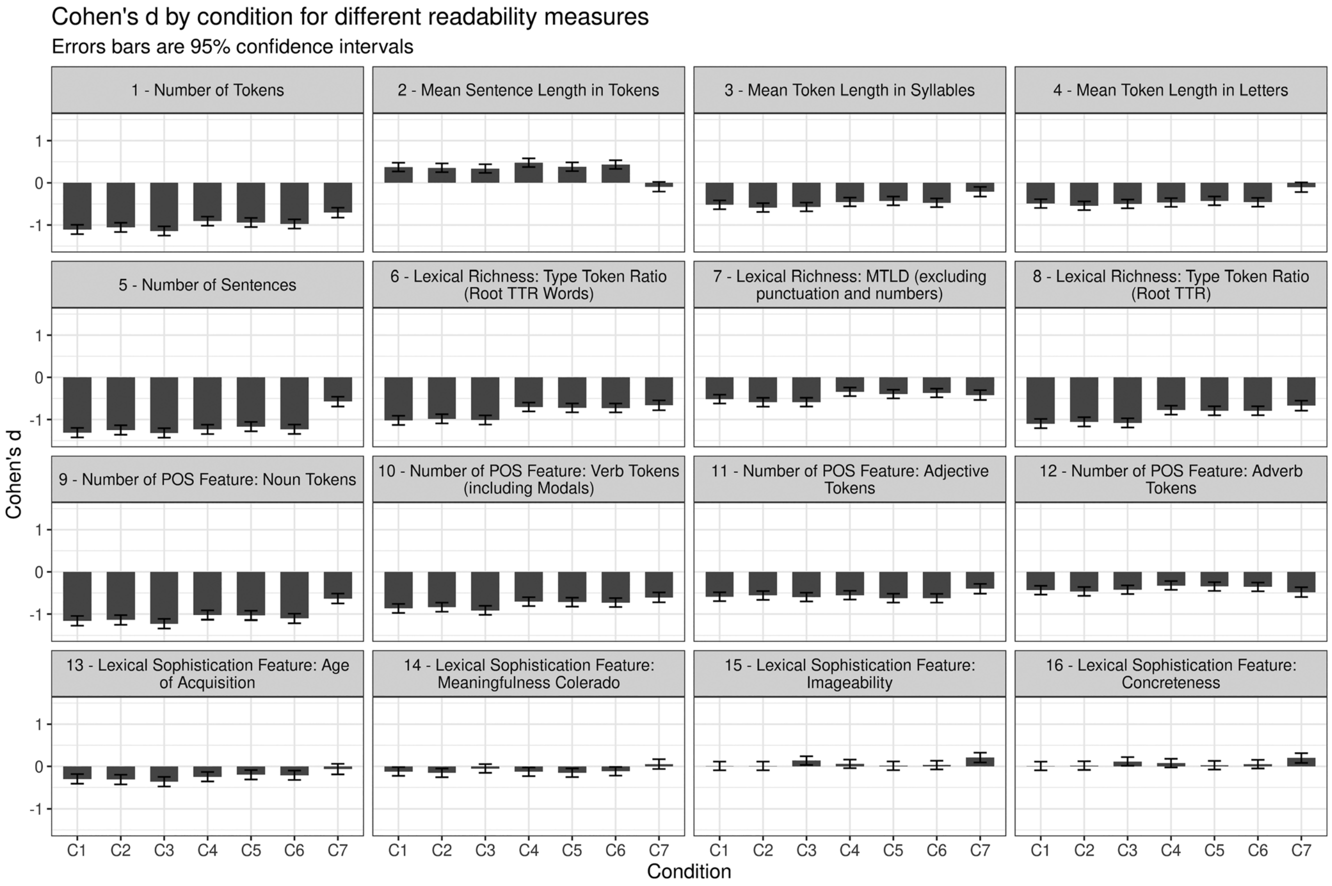

FIGURE 1 Visualization of Cohen's d and their confidence intervals for each condition and readability measure

After applying the initial set of thresholds described at the end of Section 6, we applied a filtering condition on any one of the readability metrics and randomly sampled three examples to see the kind of simplifications that GPT-3 was capable of making. Our small samples only gave us examples of what GPT-3 was capable of doing, and they did not give us an indication of what fraction of the passages were simplified meaningfully. In the table below, we show examples of simplifications found by using threshold values on the readability metrics:

Here are the key observations in the table above. We have not described examples that are not entirely meaningful.

- Number of Tokens: We can see that the last two examples found using metric are meaningful. The outputs have less words in them and they communicate the same information as the input.
- Mean Sentence Length: We can see that the first example has two sentences in the input, whilst the output breaks the second sentence of the input into two individual sentences. This results in a lower sentence length. The last sentence of the second example is simplified from "Illustrate and explain the calculation by using equations, rectangular arrays, and/or area models". to "Illustrate and explain the calculation by drawing pictures".
- Lexical Richness: MTLD: We do not find any of the simplifications useful here.
- Lexical Richness: Type Token Ratio: We can see that in the first example, the word "number" is repeated in the output whilst words "number" and "product" are used to mean the same thing. Repeating a word leads to lower TTR. We can see that for the second example, the same information is being communicated in less words in the output.
- Lexical Sophistication Feature: Age of Acquisition: The first and second examples have simpler structures in the output.

From the table above, we can see that GPT-3 is capable of providing meaningful simplifications of math word problems. The data support the H2 and show that GPT-3 AI is capable of providing meaningful simplifications of math word problems. It is important to note that the number of samples based on which we are making our conclusion is small, but they clearly show that the AI engine is capable of producing useful simplifications.

8 | DISCUSSION

Several examples from Table 10 can be given as suggestions to item authors when they are authoring the original text of the question. The filtering conditions yielded sets that were very small compared to the

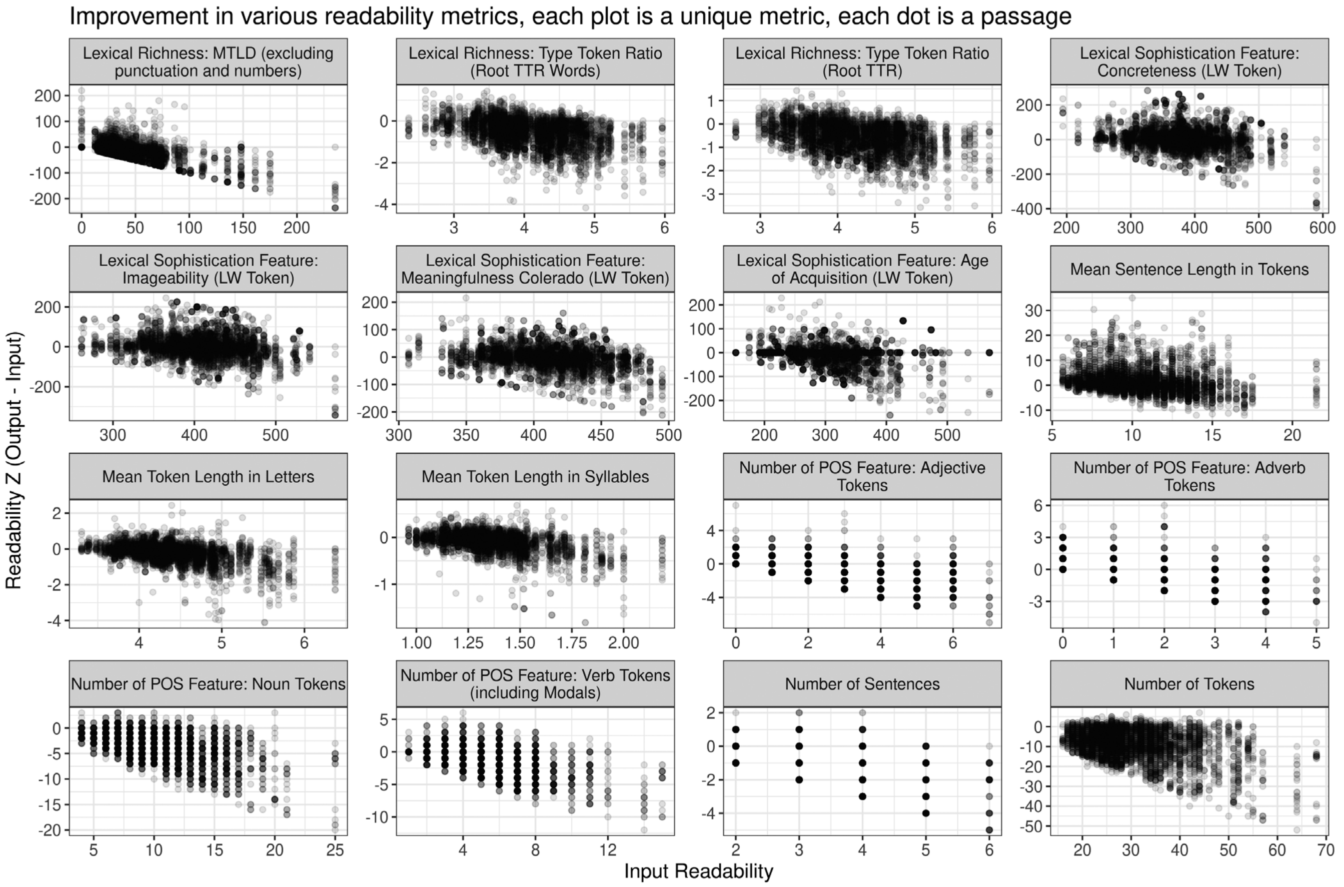

FIGURE 2 Changes observed in various readability metrics from input to output

total samples generated by the API (the filter set sizes ranged from 1% to 4%). The useful sample count in the filter sets was further lower than this. Based on these numbers, we can say that the accuracy of our method is quite low and not at a level where our method can be used in a real-world context. Our results reveal a promising capability that the AI engine has, but more work is needed to improve the rate of producing meaningful results.

We used two types of numeric metrics to identify useful simplifications - readability and text similarity. Of the two, the similarity metrics were essential in identifying useful simplifications. They acted as a noise filter and allowed us to find passages that had a minimum level of similarity with the input. This is a desired outcome when simplifying math word problems, where we would expect most of the key information in the input to be retained in the output. The median percentage of common words between input and output was 45%, telling us that more than half of the translations did not carry the majority of the original words. The differences in usefulness between the readability metrics could be due to random sampling, but it is clear that if we had not used the similarity metrics, we would have easily gotten a large number of noisy samples. The observation of the usefulness of the similarity metrics partially answers RQ1 and tells us that we can use similarity metrics to look for good simplifications. To calculate which readability metric is most helpful in finding good simplifications, we need to create a benchmark dataset of human simplifications and analyse it with the given metrics to understand the extent to which we can expect the given readability metric to change. Creating this benchmark dataset will allow us to go deeper into RQ2 and understand which metrics can identify human simplifications. Our data and analysis were not sufficient to answer RQ2 and identify which metrics are most useful.

Our method did not have a high rate of producing meaningful simplifications, but the results demonstrate the capability of the AI engine to impact the quality of assessment questions positively. Given the success rate of our method, we found the text similarity metrics to be quite helpful in filtering out the noisy simplifications. We saw that many simplifications were simply not useful and often had nothing to do with the input. Similarity metrics quickly filtered out this noise and allowed us to discover simplifications that had at least a minimum amount of similarity with the input. It is possible to improve readability of a text passage by altering its context and making it more familiar to the learner. In some cases, without changing the syntax of the text, we can simply replace nouns with alternatives and make the text more context sensitive. In such cases, even though the text may be called identical in some sense, the text similarity measures may not capture it fully. Similarity measures based on word vectors or word

TABLE 10 Simplifications found using metric thresholds

Readability metric and filtering condition	Input	Simplification by GPT-3
Number of Tokens Difference <= −5 # Passages: 140	Find whole-number quotients of whole numbers with up to four-digit dividends and two-digit divisors using strategies based on place value, the properties of operations, and/or the relationship between multiplication and division. Illustrate and explain the calculation by using equations, rectangular arrays, and/or area models.	Find whole-number quotients of whole numbers using strategies based on place value, the properties of operations, and/or the relationship between multiplication and division. Illustrate and explain the calculation by using equations, rectangular arrays, and/or area models.
	Ms. Leverenz is doing an art project with her class. She has a 3 foot piece of ribbon. If she gives each student an eighth of a foot of ribbon, will she have enough for her class of 22 students?	Ms. Leverenz has a 3 foot piece of ribbon. If she gives each student 1 inch of ribbon, will she have enough for her class of 22 students?
	Jon pours the contents of Container 1 and Container 3 above into an empty bucket. How much liquid is in the bucket after he pours the liquid?	Jon pours the contents of Container 1 and Container 3 into an empty bucket. How much liquid is in the bucket now?
Mean Sentence Length in Tokens Difference <= −2 # Passages: 46	On the number line above, use a blue coloured pencil to divide each whole into eighths, and label each fraction below the line. Refold your fraction strip from Problem 1 to help you estimate.	On the number line above, use a blue coloured pencil to divide each whole into eighths. Label each fraction below the line. Refold your fraction strip from Problem 1 to help you estimate.
	Find whole-number quotients of whole numbers with up to four-digit dividends and two-digit divisors using strategies based on place value, the properties of operations, and/or the relationship between multiplication and division. Illustrate and explain the calculation by using equations, rectangular arrays, and/or area models.	Find whole-number quotients of whole numbers with up to four-digit dividends and two-digit divisors using strategies based on place value, the properties of operations, and/or the relationship between multiplication and division. Illustrate and explain the calculation by drawing pictures.
	Step 1: Draw and shade a tape diagram of the given fraction. Step 2: Record the decomposition of the fraction in three different ways using number sentences.	Step 1: Draw and shade a tape diagram of the given fraction. Step 2: Record the decomposition of the fraction using number sentences.
Lexical Richness: MTLD (excluding punctuation and numbers) Difference <= −5 units Scale typically ranges between 0 to 100 # Passages: 178	Samantha ran 3003 m in 7 days. If she ran the same distance each day, how far did Samantha run in 3 days?	If Samantha ran the same distance each day, how far would she have run in 3 days?
	Step 1: Draw and shade a tape diagram of the given fraction. Step 2: Record the decomposition of the fraction in three different ways using number sentences.	Step 1: Draw and shade a tape diagram of the given fraction. Step 2: Record the decomposition of the fraction using number sentences.
	Mrs. Silverstein bought 3 mini cakes for a birthday party. She cuts each cake into quarters and plans to serve each guest 1 quarter of a cake. How many guests can she serve with all her cakes? Drawa picture to support your response.	Mrs. Silverstein bought 3 mini cakes for a birthday party. She cuts each cake into quarters and plans to serve each guest 1 quarter of a cake. How many guests can she serve with all her cakes? Draw a picture to support your response.
Lexical Richness: Type Token Ratio (Root TTR) Difference <= −0.25 # Passages: 144	Jack said that if you take a number and multiply it by a fraction, the product will always be smaller than what you started with. Is he correct? Why or why not? Explain your answer, and give at least two examples to support your thinking.	Jack said that if you multiply a number by a fraction, the result will always be smaller than the original number. Is he correct? Why or why not? Explain your answer, and give at least two examples to support your thinking.
	Mrs. Silverstein bought 3 mini cakes for a birthday party. She cuts each cake into quarters and plans to serve each guest 1 quarter of a cake. How many guests can she serve with all her cakes? Drawa picture to support your response.	Mrs. Silverstein bought 3 mini cakes for a birthday party. If she cuts each cake into quarters, how many guests can she serve each quarter of a cake? She can serve up to 9 guests with all her cakes.
	The teacher has 12 green stickers and 15 purple stickers. Three students are given an equal number of each colour sticker. How many green and purple stickers does each student get?	The teacher has green and purple stickers. Three students are given an equal number of green and purple stickers. How many green and purple stickers does each student get?

(Continues)

TABLE 10 (Continued)

Readability metric and filtering condition	Input	Simplification by GPT-3
Lexical Sophistication Feature: Age of Acquisition (LW Token) Difference <= −1 # Passages: 108	Use the place value chart and arrows to show how the value of each digit changes. The first one has been done for you.	Write the number 123. Draw arrows to show how the value of each digit changes. The first one has been done for you.
	A garden box has a perimeter of 27 feet. If the length is 9 feet, what is the area of the garden box?	A garden box is 9 feet long and has a perimeter of 27 feet. What is the area of the garden box?
	Find whole-number quotients of whole numbers with up to four-digit dividends and two-digit divisors using strategies based on place value, the properties of operations, and/or the relationship between multiplication and division. Illustrate and explain the calculation by using equations, rectangular arrays, and/or area models.	Find whole-number quotients of whole numbers with up to four-digit dividends and two-digit divisors using strategies based on place value, the properties of operations, and/or the relationship between multiplication and division.

overlap might find two texts different even though their content has little difference at its core.

We saw that, on average, the AI engine improved the readability of the text passages over various metrics as predicted by previous studies. GPT-3 AI has been trained on a large amount of publicly available text on the internet, and this means that it has been given a lot of text describing the concept of readability. To understand what is the source of the work the AI engine is able to do, we need to probe it with inputs that ask how the AI defines the readability. We asked the question “How do you improve the readability of a math word problem?” to the AI, and at zero temperature configuration when output is mostly the same on each run, the responses suggested breaking down bigger sentences into smaller ones, avoiding using jargon, and using visual aids. It is interesting to note here that the Few Shot Learning condition's average performance on readability metric improvement was more aligned with the directions of improvements predicted by prior studies. Although, looking at the confidence intervals of the effect sizes (in Table 9), we found that the Few Shot Learning condition's lower bound intersects with all other intervals. The prompt-based conditions did not work as predicted in two metrics (Mean Sentence Length in Tokens and Meaningfulness). This may be related to the bias that AI already has about what readability means and how it may be improved. GPT-3 model was also able togenerate the Flesch Reading Ease formula when we prompted it with a text that asked it to write down the formula. It is likely that web content on the older methods to measure readability is available in higher quantities than the more recent methods and approaches. In this case, AI's learnings may be biassed against the latest research. We believe that to improve the accuracy of our approach, the AI must be trained using several hundred training examples created by experts. Prior studies around automated simplification of educational texts have focused on making reading passages easier to read (De Belder & Moens, 2010; Rebello et al., 2019). One study by Nandhini and Balasundaram (2013) focused on improving the readability of math word problems. The task of automatically simplifying math word problems differs from other automated simplification tasks mainly with respect to the size of the input text. Typically, the input is short, which makes it difficult to use measures like cohesion to measure text readability. In our analysis, we found that the MTLD measure (Measure of Textual Lexical Diversity), which has been found to be stable for shorter texts (Zenker & Kyle, 2021), did not help us in finding good simplifications. The method that we have used in our study is markedly different from others with regard to its usability. Any individual with technical knowledge of APIs can leverage the capabilities of GPT-3 AI. If the accuracy of our approach can be improved by using training examples instead of prompts or by discovering better prompts, educational content authoring tools can embed the text simplification model into their system with a moderate amount of software development work. Figure 3 below shows how any text AI-based simplification system can be used in a real-world scenario.

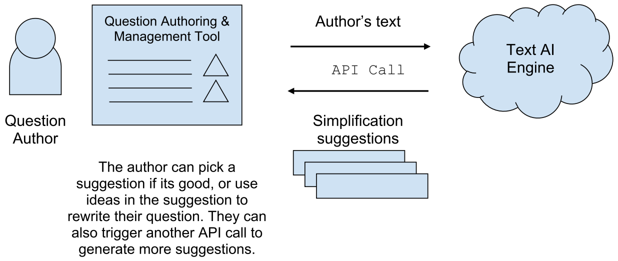
Question Authoring & Management Tool
Question Author
Author's text
API Call
Text AI Engine
Simplification suggestions
The author can pick a suggestion if its good, or use ideas in the suggestion to rewrite their question. They can also trigger another API call to generate more suggestions.

FIGURE 3 An example system showing how a text-AI model can assist question authors with item simplification suggestions

A very important thing to note here is that the text AI will only be able to provide suggestions, and we expect errors to be present. Unless the AI system demonstrates an extremely high degree of accuracy, the simplification suggestions cannot be automatically used in place of the original question text. Text AI engines generate the output text by using sampling methods. This means that unlikely words can appear in the output, and hence the output must be reviewed before it is used in a real-world context.

The key limitation of our study is the absence of a dataset that has training examples of simplifications that AI can learn from. There is no publicly available dataset that has examples of math world problem simplifications. There are several benchmark datasets for text simplification that contain pairs of difficult and simplified sentences. We were not able to use these datasets for our task due to validity concerns. For example, the Wikipedia simplification dataset has single sentence inputs and outputs, which is unlike a typical math word problem. If we validate our method with the Wikipedia dataset, the results may not generalize to datasets of math word problems. Our study also did not describe the extent to which we can expect the readability metrics to improve. This was mainly due to the absence of a benchmark dataset that contains educational content. If a dataset of several hundred math word problems and their simplifications is created by experts, it can be analysed to understand the extent to which various readability metrics can improve when we try to improve readability ourselves. These data points can be used as a threshold to identify highly relevant simplifications from the many that AI engines can generate.

9 | CONCLUSION

In conclusion, we found GPT-3 AI capable of producing meaningful simplifications of math word problems. On average, the readability metrics saw improvement in expected directions from inputs to outputs. Our methods did not yield simplifications with high accuracy. We used text similarity and readability measures to filter the noisy outputs and identify useful outputs. The text similarity metrics were particularly helpful in filtering out the noisy text. The useful simplification we found could be shown to test creators as suggestions and ideas for improvement at the time of question authoring. Our approach is straightforward to implement in real-world scenarios where question authoring happens in digital interfaces. To realize the potential of text AI in improving assessment quality, we need to create a publicly available dataset of word problem simplifications. The benchmark dataset can train the AI and make it more accurate. We believe that creating this dataset is the key next step to making more progress in the direction of automated math world problem simplification.

10 | OPEN-SOURCE DATA

To enable others to discover more interesting examples from the full set, we are open-sourcing the data from our analysis. Our data contains all input and output passages and all of the mentioned readability and text similarity metrics. Link to data: https://doi.org/10.5281/zenodo.6809166

11 | FUTURE WORK

The extension of our work will require the creation of a dataset that has several hundred examples of how word problems can be simplified. This data can fine-tune the GPT-3 AI. We explored the Few Shot capability of the model in this analysis, but GPT-3 now also provides a fine-tuning API that can create a custom model for a specific task.

GPT-3 now provides text editing capabilities, where the engine edits the provided text as per the instructions written in the prompt. This capability can be used to contextualize educational texts and make sure that the vocabulary used in the text is relevant to the students. This can enable us to potentially adapt open-source curricula to local contexts. Walkington et al. (2014) and Bernacki and Walkington (2018) showed that incorporating students' out-of-school interests in educational texts can affect their math learning. We can potentially use automated text editing to generate content with motivating elements for the students.

PEER REVIEW

The peer review history for this article is available at https://publons.com/publon/10.1111/jcal.12776.

DATA AVAILABILITY STATEMENT

Our study analyzes natural language data generated from the GPT-3 AI system. We hereby declare that no part of our manuscript is written by an AI system and none of the data used for our study was written by a human. Our data are available publicly for review at https://doi.org/10.5281/zenodo.6809166.

ORCID

Nirmal Patel https://orcid.org/0000-0003-1472-4029

REFERENCES

Al-Thanyyan, S. S., & Azmi, A. M. (2021). Automated text simplification: A survey. *ACM Computing Surveys (CSUR)*, *54*(2), 1–36.

Benjamin, R. G. (2012). Reconstructing readability: Recent developments and recommendations in the analysis of text difficulty. *Educational Psychology Review*, *24*(1), 63–88.

Bernacki, M. L., & Walkington, C. (2018). The role of situational interest in personalized learning. *Journal of Educational Psychology*, *110*(6), 864–881.

Bertram, B., & Newman, S. (1981). Why readability formulas fail (Report No. 28). Illinois University, Urbana: Center for the Study of Reading. (Eric Document Service No. ED205915).

Britton, B. K., & Gülgöz, S. (1991). Using Kintsch's computational model to improve instructional text: Effects of repairing inference calls on recall and cognitive structures. *Journal of Educational Psychology*, *83*(3), 329–345.

Chandrasekar, R., & Srinivas, B. (1997). Automatic induction of rules for text simplification. *Knowledge-Based Systems*, *10*(3), 183–190.

Chen, X., & Meurers, D. (2016, December). CTAP: A web-based tool supporting automatic complexity analysis. Proceedings of the workshop on computational linguistics for linguistic complexity (CL4LC), 113–119.

Corlatescu, D. G., Dascalu, M., & McNamara, D. S. (2021, June). Automated model of comprehension V2. 0. International Conference on Artificial Intelligence in Education, 119–123. Springer, Cham.

Crossley, S. A., Kyle, K., & McNamara, D. S. (2016). The tool for the automatic analysis of text cohesion (TAACO): Automatic assessment of local, global, and text cohesion. *Behavior Research Methods*, *48*(4), 1227–1237. https://doi.org/10.3758/s13428-015-0651-7

Cummins, D. D., Kintsch, W., Reusser, K., & Weimer, R. (1988). The role of understanding in solving word problems. *Cognitive Psychology*, *20*(4), 405–438.

Dale, E., & Chall, J. S. (1949). The concept of readability. *Elementary English*, *26*(1), 19–26.

Dale, R. (2021). GPT-3: What's it good for? *Natural Language Engineering*, *27*(1), 113–118. https://doi.org/10.1017/S1351324920000601

Davidson, A., & Kantor, R. N. (1982). On the failure of readability formulas to define readable texts: A case study from adaptations. *Reading Research Quarterly*, *17*, 187–209.

De Belder, J., & Moens, M. F. (2010). Text simplification for children. Proceedings of the SIGIR workshop on accessible search systems, 19–26. ACM, New York.

Dempster, E. R., & Reddy, V. (2007). Item readability and science achievement in TIMSS 2003 in South Africa. *Science Education*, *91*(6), 906–925.

Flesch, R. (1948). A new readability yardstick. *Journal of Applied Psychology*, *32*(3), 221–233. https://doi.org/10.1037/h0057532

François, T., & Fairon, C. (2012, July). An "AI readability" formula for French as a foreign language. Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. 466–477.

Friedrich, M. C., & Heise, E. (2019). Does the use of gender-fair language influence the comprehensibility of texts? An experiment using an authentic contract manipulating single role nouns and pronouns. *Swiss Journal of Psychology*, *78*(1–2), 51–60.

Graesser, A. C., McNamara, D. S., Louwerse, M. M., & Cai, Z. (2004). Coh-Metrix: Analysis of text on cohesion and language. *Behavior Research Methods*, *Instruments*, *& Computers*, *36*(2), 193–202.

Heilman, M., Collins-Thompson, K., & Eskenazi, M. (2008, June). An analysis of statistical models and features for reading difficulty prediction. Proceedings of the third workshop on innovative use of NLP for building educational applications, 71–79.

Hwang, W., Hajishirzi, H., Ostendorf, M., & Wu, W. (2015). Aligning sentences from standard wikipedia to simple wikipedia. Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 211–217.

Janfada, B., & Minaei-Bidgoli, B. (2020, April). A review of the most important studies on automated text simplification evaluation metrics. 2020 6th International Conference on Web Research (ICWR), 271–278. IEEE.

King, D., & Burge, B. (2015). Readability analysis of PISA 2012 mathematics, science and reading assessments.

Koedinger, K. R., & Nathan, M. J. (2004). The real story behind story problems: Effects of representations on quantitative reasoning. *The Journal of the Learning Sciences*, *13*(2), 129–164.

Kyle, K., & Crossley, S. A. (2015). Automatically assessing lexical sophistication: Indices, tools, findings, and application. *TESOL Quarterly*, *49*(4), 757–786. https://doi.org/10.1002/tesq.194

Kyle, K., Crossley, S. A., & Jarvis, S. (2021). Assessing the validity of lexical diversity indices using direct judgements. *Language Assessment Quarterly*, *18*(2), 154–170.

Lamb, J. H. (2010). Reading grade levels and mathematics assessment: An analysis of Texas mathematics assessment items and their reading difficulty. *The Mathematics Educator*, *20*(1), 22–34.

Loveless, T., Williams, V., Ball, D. L., Hoffer, T. B., Venkataraman, L., & Hedberg, E. C. (2008). Report of the subcommittee on the national survey of Algebra I teachers. Foundations for success: Report of the national mathematics advisory panel.

McNamara, D. S., & Kintsch, W. (1996). Learning from texts: Effects of prior knowledge and text coherence. *Discourse Processes*, *22*(3), 247–288.

Nandhini, K., & Balasundaram, S. R. (2012). Grade level classification of math word problems to improve readability for learning disability. 2012 IEEE International Conference on Technology Enhanced Education (ICTEE). https://doi.org/10.1109/ictee.2012.6208638

Nandhini, K., & Balasundaram, S. R. (2013). Improving readability through extractive summarization for learners with reading difficulties. *Egyptian Informatics Journal*, *14*(3), 195–204.

Nelson, J., Perfetti, C., Liben, D., & Liben, M. (2012). *Measures of text difficulty: Testing their predictive value for grade levels and student performance*. Council of Chief State School Officers.

Noonan, J. (1990). Readability problems presented by mathematics text. *Early Child Development and Care*, *54*(1), 57–81. https://doi.org/10.1080/0300443900540104

OpenAI. (n.d.-a). Examples: Summary for a 2nd grader. https://beta.openai.com/docs/examples/summary-for-a-second-grader

OpenAI. (n.d.-b). Tokenizer. https://beta.openai.com/tokenizer

Pitler, E., & Nenkova, A. (2008). In Revisiting readability: A unified framework for predicting text quality. In Proceedings of the Conference on Empirical Methods in Natural Language Processing - EMNLP '08.

Prins, E., & Ulijn, J. (1998). Linguistic and cultural factors in the readability of mathematics texts: The Whorfian hypothesis revisited with evidence from the South African context. *Journal of Research in Reading*, *21*(2), 139–159.

Rakow, S. J., & Gee, T. C. (1987). Test science, not Reading. *Science Teacher*, *54*(2), 28–31.

Rebello, B. M., Santos, G. L. D., Ávila, C. R. B. D., & Kida, A. D. S. B. (2019). Effects of syntactic simplification on reading comprehension of elementary school students. *Audiology-Communication Research*, *24*.

Sadoski, M., Goetz, E. T., & Fritz, J. B. (1993). Impact of concreteness on comprehensibility, interest, and memory for text: Implications for dual coding theory and text design. *Journal of Educational Psychology*, *85*(2), 291–304.

Sadoski, M., Goetz, E. T., & Rodriguez, M. (2000). Engaging texts: Effects of concreteness on comprehensibility, interest, and recall in four text types. *Journal of Educational Psychology*, *92*(1), 85–95.

Sawilowsky, S. S. (2009). New effect size rules of thumb. *Journal of Modern Applied Statistical Methods*, *8*(2), 26–599.

Sawyer, M. H. (1991). A review of research in revising instructional text. *Journal of Reading Behavior*, *23*(3), 307–333.

Sweller, J., Van Merrienboer, J. J., & Paas, F. G. (1998). Cognitive architecture and instructional design. *Educational Psychology Review*, *10*(3), 251–296.

Walkington, C., Clinton, V., & Shivraj, P. (2018). How readability factors are differentially associated with performance for students of different backgrounds when solving mathematics word problems. *American Educational Research Journal*, *55*(2), 362–414.

Walkington, C., Clinton, V., & Sparks, A. (2019). The effect of language modification of mathematics story problems on problem-solving in online homework. *Instructional Science*, *47*(5), 499–529.

Walkington, C., Sherman, M., & Howell, E. (2014). Personalized learning in algebra. *The Mathematics Teacher*, *108*(4), 272–279.

Zenker, F., & Kyle, K. (2021). Investigating minimum text lengths for lexical diversity indices. *Assessing Writing*, *47*, 100505.

How to cite this article: Patel, N., Nagpal, P., Shah, T., Sharma, A., Malvi, S., & Lomas, D. (2023). Improving mathematics assessment readability: Do large language models help? *Journal of Computer Assisted Learning*, 1–19. https://doi.org/10.1111/jcal.12776

APPENDIX A

Human simplifications of 10 randomly selected problems. Total of $P = 11$ simplifications (#6 was simplified twice).

#	Word problem from EngageNY math curriculum	Human simplified word problem
1	Luis uses square inch tiles to build a rectangle with a perimeter of 24 inches. Does knowing this help him find the number of rectangles he can build with an area of 24 square inches? Why or why not?	You want to count all the ways to make 24 square inch rectangles. You first make a 24 inch perimeter rectangle with 1 square inch tiles. Will this help?
2	Fill in the missing whole numbers in the boxes below the number line. Rename the whole numbers as fractions in the boxes above the number line.	Write whole numbers in the boxes below the number line. Write fractions equal to whole numbers in the boxes above the number line.
3	Compare the perimeter of your tessellation to a partner's. Whose tessellation has a greater perimeter? How do you know?	Compare your pattern with someone else's pattern. Which pattern has a longer perimeter?
4	Place the two fractions on the number line. Circle the fraction with the distance closest to 0. Then compare using >, <, or =. The first problem is done for you.	Place the two fractions on the number line. Circle the fraction closest to 0. Then compare the fractions using >, <, or =. The first problem is done for you.
5	Three rectangular prisms have a combined volume of 518 cubic feet. Prism A has one-third the volume of Prism B, and Prisms B and C have equal volume. What is the volume of each prism?	Total volume of three prisms is 518 cubic feet. Prism A has one-third the volume of Prism B. Prisms B and C have equal volume. What is the volume of each prism?

APPENDIX B

Regular expression rules used to remove unformatted passages from the EngageNY data sample.

<table>
<tr><th>RegEx (used in stringr R package)</th><th>Description</th></tr>
<tr><td>\{|\}</td><td>Curly braces and pipes</td></tr>
<tr><td>\|</td><td>Pipes</td></tr>
<tr><td>[0-9]+\s+[0-9]+\s+[0-9]+</td><td>Three numbers with spaces between them</td></tr>
<tr><td>[[:punct:]] [[:punct:]]</td><td>Two punctuations with a space between them</td></tr>
<tr><td>[[:punct:]]{2}</td><td>Two consecutive punctuations</td></tr>
<tr><td>[[:punct:]]{3}</td><td>Three consecutive punctuations</td></tr>
<tr><td>[a-z]; [a-z]</td><td>Semicolon with spaces and characters around it</td></tr>
<tr><td>\. \.</td><td>Two full stops</td></tr>
<tr><td>[a-zA-Z] = [a-zA-Z]</td><td>Equal to sign surrounded by characters</td></tr>
<tr><td>;</td><td>Semicolon surrounded by spaces</td></tr>
<tr><td>NOT([\.?!]$)</td><td>Excerpt has to end with a full stop, question mark, or an exclamation sign, if not, it is removed</td></tr>
<tr><td>[1-5]\.[0-9A-Z]+\.[0-9]+</td><td>Common Core Standard codes</td></tr>
<tr><td>[0-9][a-zA-Z][0-9]</td><td>Number-character-number</td></tr>
<tr><td>[0-9][A-Za-z]</td><td>Number followed by a character without a space</td></tr>
<tr><td>[A-Za-z][0-9]</td><td>Character followed by a number without a space</td></tr>
<tr><td>[^0-9A-Za-z]</td><td>Anything non alpha-numeric with spaces around it</td></tr>
<tr><td>=[0-9A-Za-z]</td><td>No character after the equal to sign</td></tr>
</table>