

AI Assistant for Editing Mathematical Expressions in Polish*

Agnieszka BIER and Zdzislaw SROCZYNSKI

Faculty of Applied Mathematics, Silesian University of Technology, Gliwice, Poland

Correspondence should be addressed to: Agnieszka BIER, agnieszka.bier@polsl.pl

* Presented at the 44th IBIMA International Conference, 27-28 November 2024 Granada, Spain

Abstract

In this paper we present a brief study on the capability of LLM to edit mathematical expressions by means of natural language instructions. Mathematical notation used in the study is encoded in the LaTeX-based EQED format, developed by the authors within the original math editing tool: *Equation wizard* (Polish “*Edytor wzorów*”) (Bier and Sroczyński 2015, 2019). We evaluate the repeatability and precision of EQED code alterations made by GPT-4o in reference to natural language instructions on required formula edition. For the three distinguished classes of editing commands: inserting, deleting, or replacing parts of mathematical expressions, we run a series of experiments and analyze the produced EQED code and its compilation obtained by prompting the LLM. We demonstrate the consistency and repeatability of LLM output based on the test requests.

Keywords: math notation edition, LaTeX mathematical notation, LLM, GPT

Introduction

Mathematical expressions are the key ingredient in many scientific and educational textbooks and papers. It creates the need for efficient and reliable editing tools that are capable of both encoding and navigating within the formulas. This knowledge recording function is essential not only in education of visually impaired or physically disabled students, who require verbalization of mathematical content for math-to-speech or speech-to-math interfaces (Abreu et al 2019, Attanayake et al 2015, Maćkowski et al 2023, Mejía et al 2021), but also has broader potential applications. For instance, verbalizing mathematical notation can be utilized for searching and indexing formulas, as well as for detecting plagiarism in academic content (Bier and Sroczyński 2018, 2021, 2022). Our primary motivation in this study is providing the most intuitive and reliable tool for editing mathematics. Hence, we considered the application of large language models (LLMs) as an intermediary agent to capture and process natural language commands for editing mathematical content.

The symbolic nature of mathematical notation as well as the graphical inter-dependance of used symbols imposes specific challenges to the notation syntax (Cuartero-Olivera 2012, Feichter and Schlippe 2024). One of the most widely used systems is LaTeX (Thinn 2020), which is based on a set of commands enabling navigation within the expression. In our previous works, we used LaTeX as the core of our original math encoding format, EQED. It employs delimiters or environments for precise encoding of components of the formula such as brackets, exponents, numerators, denominators etc. Such syntax imposes difficulty in edition and alterations of EQED encoding. Adding,

Cite this Article as: Agnieszka BIER and Zdzislaw SROCZYNSKI, Vol. 2024 (22) “AI Assistant for Editing Mathematical Expressions in Polish” Communications of International Proceedings, Vol. 2024 (22), Article ID 4447424, <https://doi.org/10.5171/2024.4447424>

removing or replacing parts in the existing formulas require precise indication of required change and its context. Our original solution employs a set of pre-defined navigation commands that allow to interpret the intent of the user (Bier and Sroczyński 2020, 2023). Although precise, this approach does not provide high level user experience. The commands and syntax required for effective performance are not natural and need to be learned in order to be used for the interface. On the other hand, nowadays we observe the rapid growth in applications of Large Language Models (LLMs) as a user-friendly interface of a variety of solutions. Thus, an interesting question arises, whether such models can be successfully used for adjusting the encoded formulas directly in the syntax with natural language commands. Two aspects are of particular interest in our experiments: the capability of LLM to properly navigate within the formula and the ability to decode and encode formulas in the original EQED format, given that the model could rely only on examples and request provided in the prompt. This challenge becomes especially important when working with the Polish language, where command dictionaries are not always effective due to the complexity introduced by inflections, which can significantly alter the phrasing of user commands.

The idea of applying LLM for code alterations is already present and explored in many areas, especially the software development (Jiang 2023, Schumann et al 2024). The specialized AI agents are capable of explaining, generating, editing – including code refactor and adding unit tests, the portions of source code in various programming languages based on the descriptive prompt. One of the most well-known implementations of such kind of AI agent is GitHub Copilot (Copilot 2024), which has demonstrated significant capabilities in assisting developers by leveraging natural language inputs to produce and modify code efficiently (Coldwind 2023). In these cases, however, the models are equipped with large knowledgebase with numerous examples as the use case is widely present in public domain. With the original EQED format, however, the approach is fundamentally distinct: the entire context and knowledge base required for the model to perform the task must be explicitly provided within the prompt.

Architecture

The *Equation wizard* (Polish “*Edytor wzorów*”) Windows app offers many different ways for user interaction, which includes editing through a structured interface of menus and specially designed panels with buttons, dictation based on dictionary-defined rules, as well as specific editing voice commands that require precise phrasing and a strict syntax. Actual dictation can also be performed via a mobile assistant application, which provides the voice input and additionally presents a real-time visual representation of the processed formula on the mobile screen. It is worth to note that with the release of Windows 11, direct dictation in Polish is now available in any edit field, enhancing UX for Polish-speaking users.

The editor has been enhanced with UI components that provide a user interface for the underlying assistant functionality: command entry text field and an activation button. Some additional number fields related to API parameters are also present, and some are left for debugging purposes and will be eventually hidden from the end user.

The assistant — simply named “Helper” (“Pomocnik” in the Polish version) — executes commands provided by the user in the natural language. These commands are transmitted as a specially formatted prompt to the API of a selected generative AI model, currently GPT-4o by OpenAI. The prompt includes a current (source) representation of the formula, which is encoded in an internal EQED notation. This notation can be translated into standard LaTeX using a set of predefined macro definitions, which are included in the prompt for accurate conversion. The prompt used for communication with the AI API consists of a few particular sections:

```
Notacja EQED służy do opisu wzorów matematycznych.
Notację EQED definiujemy poprzez zestaw makr w LaTeX:
\def\ds{\displaystyle}
\def\EQEDhi#1#2{{#2}^{#1}}
\def\EQEDlo#1#2{{#2}_{#1}}
\def\EQEDboth#1#2#3{{#3}_{#1}^{#2}}
\def\EQEDint#1#2#3{\int\limits_{#1}^{#2}{#3}}
\def\EQEDsum#1#2#3{\sum\limits_{#1}^{#2}{#3}}
\def\EQEDprod#1#2#3{\prod\limits_{#1}^{#2}{#3}}
\def\EQEDbrackets#1#2#3{#1 {#2} #3}
\def\EQEDroot#1#2{\sqrt[#1]{#2}}
\def\EQEDchoose#1#2{{#1 \choose #2}}
```

```

\def\EQEDlim#1#2#3{\ds #1_{#2}{#3}}
\def\EQEDover#1#2{\begin{array}{c} {#1}\\ {#2} \end{array}}
\def\EQEDplain#1{#1}
\def\EQEDnplain#1{#1}
\def\o{o}
\def\X{X}
\def\P{P}
\def\notsubset{\not\subset}
\def\notin{\not\in}
\def\degree{^\circ}
\def\EQEDfrac#1#2{\ds\frac{#1}{#2}}

```

Pozostałe rozkazy zaczynające się od \ ale bez przedrostka EQED nie podlegają tłumaczeniu i mają takie same znaczenie jak w czystym LaTeX.

Dany jest źródłowy wzór w notacji EQED:
Początek wzoru źródłowego.

Koniec wzoru źródłowego.

W źródłowym wzorze zapisanym w notacji EQED zignoruj odstępy.
Następnie wykonaj na nim polecenie, które podam na końcu. Wykonując polecenie, staraj się zachować standardowe reguły matematyczne.
Zwróć wyników wzór w notacji EQED bez dodatkowych komentarzy.
Notacja EQED powinna być zastosowana do wszystkich elementów wzoru, również tych zmienionych lub dodanych. Każdą liczbę należy zapisać za pomocą rozkazu \EQEDnplain. Każdy symbol matematyczny będący literą, taki jak np. a, b, c, x, y należy zapisać za pomocą rozkazu \EQEDplain. Rozkazy \EQEDnplain i \EQEDplain nie mogą zawierać innych rozkazów. Jeżeli symbol matematyczny jest zapisany z przedrostkiem \ to nie należy go otaczać za pomocą \EQEDplain, dotyczy to greki i symboli specjalnych.

Nie włączaj do odpowiedzi makr definiujących notację EQED.
Nie oznaczaj w żaden sposób kodu w notacji EQED w odpowiedzi.
Teraz wykonaj następujące polecenie:
^^^

This prompt is designed to describe the conversion and processing of mathematical formulas using EQED notation while interacting with the AI model. First, a set of LaTeX macro definitions is provided to define the EQED notation. The section labeled with "###" is a placeholder for the actual source mathematical formula, written in EQED notation. The prompt instructs the AI to ignore spaces within the source formula to avoid formatting inconsistencies. The prompt includes also specific rules for handling numbers and variables: numbers should be wrapped in the command \EQEDnplain, mathematical symbols (e.g., letters such as a, b, x, y) should be enclosed using \EQEDplain, symbols with a \ prefix (like Greek letters or special symbols) are to be left unchanged and not wrapped in the EQED commands.

The section marked as "^^^" is a placeholder where the user's command is put. User's command can contain directives for editing parts of the expression, converting it to a different format, or performing mathematical transformations. Next, the prompt guides the AI to interpret the user's command while adhering to standard mathematical conventions. Finally, after the AI processes the source formula and executes the user's command, it is instructed to return the resulting formula in EQED notation. The response should exclude any comments or macro definitions, and contain just pure mathematical content. The prompt emphasizes that all elements, including newly generated or modified parts of the formula, must comply with the EQED notation.

As a result, we receive a formula output in EQED notation, which is automatically inserted into the editing buffer/window within the "Equation wizard" application. Users can freely combine various editing methods: making manual adjustments via the menu-and-panels interface, issuing commands based on a predefined dictionary, or performing a conversation with the AI Assistant. If needed, the processed and output formats of the formula can be easily switched from EQED to standard LaTeX by making a minor modification to the query. This flexibility is advantageous, as the LaTeX language has extensive documentation and many more resources compared to the specialized EQED format, meaning that certain queries may yield more precise results in LaTeX. It is worth noting, that "Equation wizard" can import pure LaTeX code into its internal format and visualize the formula regardless of

the input notation. Supplementary prompt ensuring the conversion of the source formula to LaTeX, execution of the user command, and returning the result also in LaTeX format is defined as:

skonwertuj wzór źródłowy na LaTeX, <actual user command in Polish>, następnie zwróć czysty LaTeX, bez rozkazów EQED, nie używaj znaczników otoczenia matematycznego, nie wykonuj przekształceń i nie komentuj działań, zwróć tylko ostateczną treść wzoru

This extension of the prompt provides switching the formula processing from EQED notation to standard, pure LaTeX format. Its primary goal is to process the source expression into pure LaTeX and handle user instructions more flexibly in the main step. First, the AI is instructed to translate the provided source formula, given in EQED notation, into standard LaTeX. This ensures compatibility with a broader range of mathematical documentation and tools which support LaTeX natively. When the formula is converted, the AI processes the specified user command (provided in Polish) adjusting, modifying or refining the formula according to the user's directives. The prompt asks the AI to return also the final output in pure LaTeX format, explicitly omitting any EQED-specific commands and excluding mathematical environments.

During the conducted experiments, it was found that an approach involving an additional conversion to pure LaTeX format gave slightly better results for some complex mathematical expressions. However, since this introduces an extra layer of abstraction and, on the other hand LaTeX import functionality in our editor is still experimental, we aimed to verify the possibility of processing the entire new format (i.e. EQED) directly within the prompt content. Therefore, we plan to use mainly EQED notation, which is native to our formula editor, as long as it is fruitful.

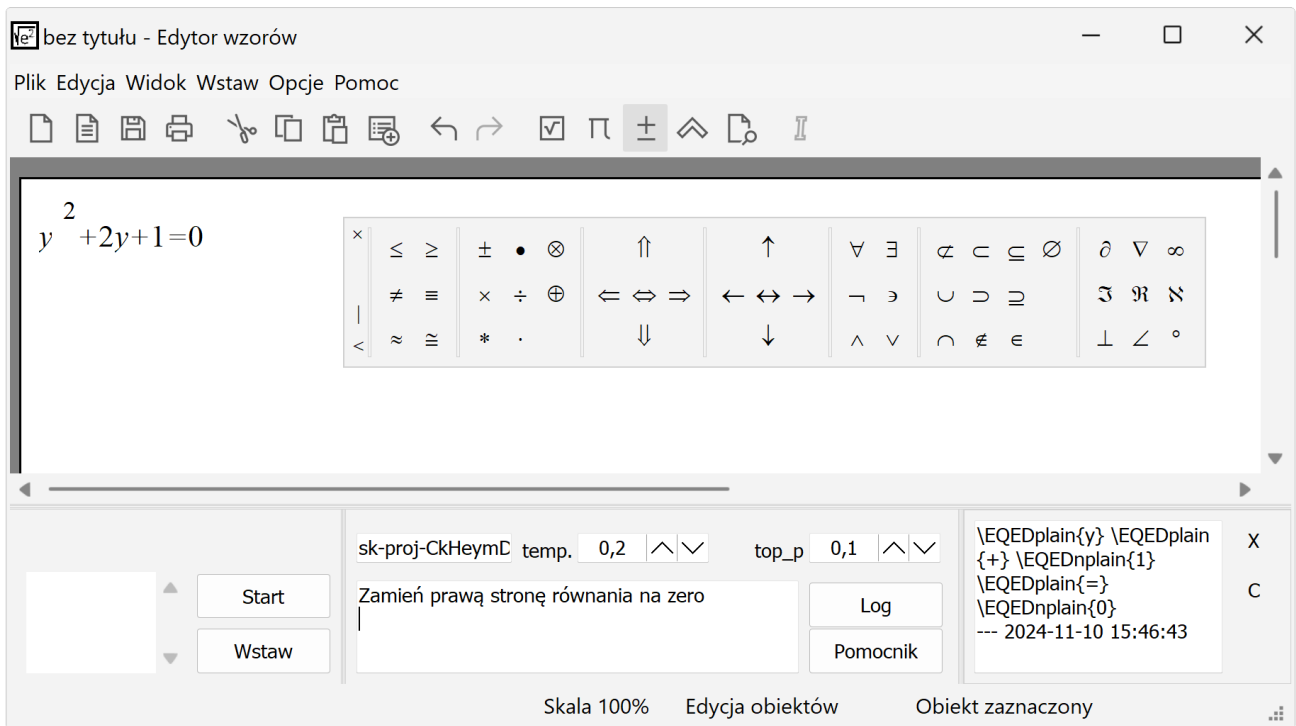


Fig 1. Graphic user interface of the “Equation wizard” editor, with AI Assistant (“Pomocnik”) visible in the middle of the lower panel. Parameters “temperature” and “top_p” setup with default values.

Our solution of the interactive assistant interface within the *Formula wizard* is implemented with the use of the REST API provided by OpenAI (<https://api.openai.com/v1/chat/completions>). When making a request via the POST method, in addition to the query content (messages), several additional parameters are passed, such as `model`, `max_tokens`, `temperature`, `top_p`, `frequency_penalty`, and `presence_penalty`. These parameters allow to fine tune the assistant responses accordingly to the user’s needs according to the following specification:

- **model**: specifies which model version to use for the query, changing response quality and computational cost, due to that, we used gpt-4o model in current experiments,
- **max_tokens**: defines the maximum number of tokens (words or parts of words) the response can contain, this way controlling its length (set to default 2048),
- **temperature**: adjusts the randomness of the generated responses - higher values result in more diverse outputs, while lower values (closer to 0) produce more focused and deterministic responses, we default it to 0.2,
- **top_p**: also known as nucleus sampling, limits the model to consider only the most likely tokens until their cumulative probability reaches **top_p**, lower value narrows down the range of potential responses, increasing coherence, we default it to 0.1,
- **frequency_penalty**: reduces the likelihood of repeating tokens that have already appeared in the conversation, promoting more varied outputs, (set to default 0),
- **presence_penalty**: encourages the inclusion of new topics by increasing the probability of using tokens that haven't been used previously, (set to default 0).

Currently, there is a UI in the editor which allows to adjust the **temperature** and **top_p** parameters. However, in future experiments, we plan to enable also dynamic model switching, even during the editing process of a single formula.

The interaction with the OpenAI API is performed using asynchronous requests, which ensures that the editor remains responsive even during extended processing times. In fact, the actual, relatively quick response times (below 2 sec.) from the OpenAI server ensured the efficient, interactive use of the editor during most of conducted experiments. This setup provides an almost real-time user experience, minimizing interruptions and facilitating a seamless workflow.

When a response is received from the API, the resulting formula automatically replaces the current version in the editor window. This replacement is seamlessly integrated into undo/redo stack of the editor application. This way users can effortlessly roll back changes if needed using standard UI/UX. As a result, users can easily revert to previous versions and refine their prompts to achieve better outcomes. This non-destructive approach encourages iterative experimentation, ensuring that all AI-assisted modifications are easily reversible, thereby preserving the integrity of the content.

Methodology

In our experiment we focused on basic editing tasks for altering the EQED encoded formulas. Our goal was to assess the repeatability and reliability aspects of the output provided by the tested LLM model. For all tests we have used GPT-4o.

The tests included a set of math editing tasks, categorized into the following types:

1. Insertion (I) – adding to an existing formula a new symbol in a designated place
2. Deletion (D) – removing from the formula indicated parts including all relevant syntax related commands
3. Replacement (R) – replacing a part of an existing expression with new symbols or subexpression.

The tasks were formulated in a natural language, that involves verbalization of the mathematical content. Each particular prompt contained instructions in Polish natural language on the expected formula modification. The tasks were performed in sequence. The modifications were applied to the formula obtained in the previous step.

Table 1 Exemplary prompts for editing EQED encoded equation.

Task no	Prompt in Polish (translation in English)	Type of editing command (I/D/R)	EQED encoding	Compiled output (visualized formula – actual screenshots from <i>Equation wizard</i> editor)	Correct? (Y/N)
1.1	zapisz równanie x do kwadratu plus dwa x równa się minus jeden (create equation x squared plus two x equals minus one)	I	$\backslash\text{EQEDplain}\{x\}\backslash\text{EQEDhi}\{\backslash\text{EQEDnplain}\{2\}\}\{\}\backslash\text{EQEDplain}\{+\}\backslash\text{EQEDnplain}\{2\}\backslash\text{EQEDplain}\{x\}\backslash\text{EQEDplain}\{=\}\backslash\text{EQEDplain}\{-\}\backslash\text{EQEDnplain}\{1\}$	$x^2 + 2x = -1$	Y
1.2	zamień x na y (substitute x with y)	R	$\backslash\text{EQEDplain}\{y\}\backslash\text{EQEDhi}\{\backslash\text{EQEDnplain}\{2\}\}\{\}\backslash\text{EQEDplain}\{+\}\backslash\text{EQEDnplain}\{2\}\backslash\text{EQEDplain}\{y\}\backslash\text{EQEDplain}\{=\}\backslash\text{EQEDplain}\{-\}\backslash\text{EQEDnplain}\{1\}$	$y^2 + 2y = -1$	Y
1.3a	Dodaj obustronnie jeden (add one sidewise)	I	$\backslash\text{EQEDplain}\{y\}\backslash\text{EQEDhi}\{\backslash\text{EQEDnplain}\{2\}\}\{\}\backslash\text{EQEDplain}\{+\}\backslash\text{EQEDnplain}\{2\}\backslash\text{EQEDplain}\{y\}\backslash\text{EQEDplain}\{=\}\backslash\text{EQEDplain}\{-\}\backslash\text{EQEDnplain}\{1\}\backslash\text{EQEDplain}\{+\}\backslash\text{EQEDnplain}\{1\}$	$y^2 + 2y = -1 + 1$	N
1.3b	Dodaj do obu stron równania jeden (add one to each side of the equation)	I	$\backslash\text{EQEDplain}\{y\}\backslash\text{EQEDhi}\{\backslash\text{EQEDnplain}\{2\}\}\{\}\backslash\text{EQEDplain}\{+\}\backslash\text{EQEDnplain}\{2\}\backslash\text{EQEDplain}\{y\}\backslash\text{EQEDplain}\{+\}\backslash\text{EQEDnplain}\{1\}\backslash\text{EQEDplain}\{=\}\backslash\text{EQEDplain}\{-\}\backslash\text{EQEDnplain}\{1\}\backslash\text{EQEDplain}\{+\}\backslash\text{EQEDnplain}\{1\}$	$y^2 + 2y + 1 = -1 + 1$	Y
1.4	Zamień prawą stronę równania na zero (replace right side with zero)	R	$\backslash\text{EQEDplain}\{y\}\backslash\text{EQEDhi}\{\backslash\text{EQEDnplain}\{2\}\}\{\}\backslash\text{EQEDplain}\{+\}\backslash\text{EQEDnplain}\{2\}\backslash\text{EQEDplain}\{y\}\backslash\text{EQEDplain}\{+\}\backslash\text{EQEDnplain}\{1\}\backslash\text{EQEDplain}\{=\}\backslash\text{EQEDnplain}\{0\}$	$y^2 + 2y + 1 = 0$	Y
1.5	Zamień lewą stronę równania na y plus jeden (replace left side with y plus one)	R	$\backslash\text{EQEDplain}\{y\}\backslash\text{EQEDplain}\{+\}\backslash\text{EQEDnplain}\{1\}\backslash\text{EQEDplain}\{=\}\backslash\text{EQEDnplain}\{0\}$	$y + 1 = 0$	Y
1.6	podnieś lewą stronę równania do kwadratu (write left side of the equation in the power of two)	I	$\backslash\text{EQEDplain}\{\{\}\}\backslash\text{EQEDplain}\{y\}\backslash\text{EQEDplain}\{+\}\backslash\text{EQEDnplain}\{1\}\backslash\text{EQEDplain}\{\}\}\{\}\backslash\text{EQEDhi}\{\{2\}\}\{\}\backslash\text{EQEDplain}\{=\}\backslash\text{EQEDnplain}\{0\}$	$(y + 1)^2 = 0$	Y

1.7	zapisz każdą stronę równania pod pierwiastkiem (write both sides of the equation under square roots)	I	$\sqrt[2]{(y+1)^2} = \sqrt[2]{0}$	Y
1.8	po lewej stronie równania usuń pierwiastek oraz wykładnik 2 (remove square root and exponent on the lefthand side)	D	$y+1 = \sqrt[2]{0}$	Y
1.9	prawą stronę równania zastąp zerem (replace right side of the equation with zero)	R	$y+1=0$	Y
1.10	usuń jedynkę po lewej stronie równania i odejmij ją od prawej strony równania (remove plus one on the left side of equation and subtract it from the right side)	D	$y=0-1$	Y
2.1	Zapisz ułamek pozostawiając licznik i mianownik pusty (Write a fraction, leaving numerator and denominator empty)	I	-	Y
2.2a	W liczniku zapisz sumę kwadratów x, y i z (put a sum of three squares in the numerator, of x, y and z)	I	Not compilable	N

2.2b	<p>W liczniku zapisz sumę x do kwadratu plus y do kwadratu plus z do kwadratu</p> <p>(In the numerator put a sum of x squared plus y squared plus z squared)</p>	I	$\frac{\text{\EQEDplain}\{x\}^{\text{\EQEDplain}\{2\}}+\text{\EQEDplain}\{y\}^{\text{\EQEDplain}\{2\}}+\text{\EQEDplain}\{z\}^{\text{\EQEDplain}\{2\}}}{\text{\EQEDplain}\{x\}}$ $\frac{\text{\EQEDsum}\{\text{\EQEDplain}\{x\}=\text{\EQEDplain}\{1\}\}\{\text{\EQEDplain}\{3\}\}\{\text{\EQEDplain}\{x\}^{\text{\EQEDplain}\{2\}}\}\{\text{\EQEDplain}\{x\}^{\text{\EQEDplain}\{2\}}\}}{\text{\EQEDplain}\{x\}}$	$\frac{xyz}{x^2+y^2+z^2}$ $\sum_{x=1}^3 x$	N
2.2 c	<p>W liczniku wpisz wyrażenie x plus y</p> <p>(in the numerator write the expression x plus y)</p>	I	$\frac{\text{\EQEDbrackets}\{\text{\EQEDplain}\{x\}+\text{\EQEDplain}\{y\}\}}{\text{\EQEDplain}\{x\}+\text{\EQEDplain}\{y\}}$ $\frac{\text{\EQEDplain}\{x\}}{\text{\EQEDplain}\{x\}+\text{\EQEDplain}\{y\}}$	$\frac{x+y}{x+y}$	Y
2.3	<p>W mianowniku wpisz wyrażenie x minus y</p> <p>(in the denominator write the expression x minus y)</p>	I	$\frac{\text{\EQEDplain}\{x\}}{\text{\EQEDplain}\{x\}-\text{\EQEDplain}\{y\}}$	$\frac{x+y}{x-y}$	Y
2.4	<p>dodaj jeden do licznika i do mianownika</p> <p>(add one to the numerator and to the denominator)</p>	I	$\frac{\text{\EQEDbrackets}\{\text{\EQEDplain}\{x\}+\text{\EQEDplain}\{y\}+\text{\EQEDplain}\{1\}\}}{\text{\EQEDplain}\{x\}-\text{\EQEDplain}\{y\}+\text{\EQEDplain}\{1\}}$	$\frac{x+y+1}{x-y+1}$	Y
2.5	<p>Usuń x z licznika i z mianownika</p> <p>(Remove x from numerator and denominator)</p>	D	$\frac{\text{\EQEDbrackets}\{\text{\EQEDplain}\{y\}+\text{\EQEDplain}\{1\}\}}{\text{\EQEDplain}\{y\}-\text{\EQEDplain}\{1\}}$	$\frac{y+1}{-y+1}$	Y

2.6	Usuń mianownik	D	$\left[\frac{y}{y+1} \right]$	$y+1$	Y
2.7	zastąp jedynkę wyrażeniem x do kwadratu	R	$\left[y \right]$	$y+x^2$	Y
2.8	dopisz w mianowniku trzy	I	$\frac{y+x^2}{3}$	$\frac{y+x^2}{3}$	Y
2.9	zamień potęgę 2 na 3	R	$\frac{y+x^2}{3}$	$\frac{y+x^3}{3}$	Y
2.10	Usuń potęgę	D	$\frac{y+x^2}{3}$	$\frac{y+x}{3}$	Y

Reliability of obtained editing commands

The overall performance of the tested model is satisfactory. It managed to successfully complete all test cases for Deletion and Replacement, struggling only with some of the Insertion type tasks. Detailed observations on the reliability of the AI assistant with respect to the type of editing commands are listed below.

Deletions: The test involved removal of different parts of the formula. For equation edition, the model was able to correctly identify and edit the sides of equation according to the provided prompt. Furthermore, it was able to localize and remove exponents and roots, as well as remove the fraction structure after deleting solely the denominator.

Replacements: The test proved the capability of the model to localize and replace selected elements of mathematical formulas. It was equally successful in editing single symbols as well as structural parts such as exponents or entire sides of the equations.

Insertions: This type of editing command appears to be the most heavily used and has revealed some weaknesses of the model. In particular, the model failed to consistently insert new elements into existing structured formulas, such

as exponents or fractions. In some test cases it was successful, while in others – despite similar syntax of the prompt – it was unable to properly encode the inserted elements in the EQED commands. The failures involved however compound functions and sidewise operations on the equation and could be resolved by more granular sequence of editing commands (compare 1.3a vs 1.3b and 2.2x vs 2.3-2.4).

Repeatability of results

We have repeated the tests for three rounds, prompting the assistant with the same sequence of requests. Setting experimentally the configurable parameters: temperature to 0.0 and top_p = 0.1 we achieved the level of repeatability of results at 100%, i.e. prompting the model three times with the same request, it produced the same output in every iteration. This is expected behavior in the case of mathematical formulas, which are naturally restricted by their mathematical sense and therefore require high precision in editing.

Summary

GPT-4o performs the majority of our test tasks without a single error. It recognizes adequate parts of the EQED code and edits accordingly to the prompt. It proved to be a convenient interface for editing mathematical content, regardless of the type of edition the user requests: insertion, replacement or deletion.

In the future research, we intend to explore more intricate scenarios that include dynamic parameter adjustments throughout editing sessions. This will involve assessing the effectiveness of various AI models, including older or simplified versions, with a focus on minimizing processing costs while maintaining performance. Moreover, we plan to compare the processing of mathematical content encoded in pure LaTeX versus that in the EQED format. The results should help to answer the question which approach is more accurate and practical for various use cases, particularly in terms of processing speed and resource efficiency.

References

- Abreu, S., Silva, Anjos, G., Guedes, H., Prietch, S., Cardoso, P. and Freire, A., (2019), “Usability evaluation of a resource to read mathematical formulae in a screen reader for people with visual disabilities”, Proceedings of the 18th Brazilian Symposium on Human Factors in Computing Systems, 2019, 1–11.
- Attanayake D., Denholm-Price J., Hunter G., Pflugel E., Wigmore A. (2015), “Speech interfaces for mathematics: Opportunities and limitations for visually impaired learners”, IMA International Conference on Barriers and Enablers to Learning Maths: Enhancing Learning and Teaching for All Learners, 1–8.
- Bier, A. and Sroczyński, Z. (2015), “Adaptive math-to-speech interface”, in Proceedings of the Multimedia, Interaction, Design and Innovation, ser. MIDI '15. New York, NY, USA: ACM, 2015, pp. 7:1–7:9. <http://doi.acm.org/10.1145/2814464.2814471>
- Bier, A. and Sroczyński, Z. (2018), “Towards semantic search for mathematical notation”, Federated Conference on Computer Science and Information Systems (FedCSIS), 2018, 465–469.
- Bier, A. and Sroczyński, Z. (2019), “Rule based intelligent system verbalizing mathematical notation,” *Multimedia Tools and Applications*, 78 (19), 28 089–28 110.
- Bier A., Sroczyński Z. (2020), “Rule-based intelligent system for dictating mathematical notation in Polish”, Mauri J.M. (ed.), ACHI 2020: The Thirteenth International Conference on Advances in Computer-Human Interactions, Valencia, Spain, 21-25 November 2020, ISBN 978-1-61208-761-0, 352-356.
- Bier, A. and Sroczyński, Z. (2021), “Reliability assessment of the automatic plagiarism detection system for various editing patterns in documents containing complex mathematical notation”, *Journal of Physics: Conference Series*, 1828 (1), 012109.
- Bier, A. and Sroczyński, Z. (2022), “Comparison of complex mathematical notation and applications for searching and plagiarism detection”, in: Soliman Khalid S. (red.): Proceedings of the 40th International Business Information Management Association Computer Science Conference (IBIMA), 23-24 November 2022, Sevilla, Spain. Theory and practice in modern computing: vision 2025 during global crisis, Proceedings of the International Business Information Management Association Conference, 2022, International Business Information Management Association, 476 s., ISBN 979-8-9867719-1-5, pp. 189-194.

- Bier, A. and Sroczyński, Z. (2023), “Expanding the capabilities of voice user interface for math formula editor through interactive commands”, in: Soliman Khalid S. (red.): Proceedings of the 41st International Business Information Management Association Computer Science Conference (IBIMA), 26-27 June 2023, Granada, Spain. Theory and practice in modern computing: vision 2025 during global crisis, Proceedings of the International Business Information Management Association Conference, 2023, International Business Information Management Association, 1878 s., ISBN 979-8-9867719-2-2, pp. 129-134.
- Copilot (2024), https://code.visualstudio.com/docs/copilot/overview#_use-cases-for-github-copilot-in-vs-code [accessed online 10.11.2024r.]
- Coldwind G. (2023), “The exam for the machine: LLMs vs programming”, (in Polish: “Egzamin dla maszyny: LLMy vs programowanie”), *Magazyn Programista* 5/2023.
- Cuartero-Olivera J., Hunter G., Perez-Navarro A. (2012), “Reading and writing mathematical notation in e-learning environments”, *eLearn Center Research Paper Series*, no. 4, 11–20.
- Feichter, C., Schlippe, T. (2024), “Investigating Models for the Transcription of Mathematical Formulas in Images”. *Applied Sciences*, 14(3), 1140.
- Jiang, C. (2023), “Investigation on the Application of Artificial Intelligence Large Language Model in Translation Tasks”. In *2023 7th International Seminar on Education, Management and Social Sciences (ISEMSS 2023)* (pp. 1341-1351). Atlantis Press.
- Maćkowski, M., Kawulok, M., Brzoza, P., & Spinczyk, D. (2023), “Methods and tools supporting the learning and teaching of mathematics dedicated to students with blindness” *Applied Sciences*, 13(12), 7240.
- Mejía, P., Martini, L., Grijalva, F., Larco, J., Rodriguez, J. (2021). “A Survey on Mathematical Software Tools for Visually Impaired Persons: A Practical Perspective.” *IEEE Access*, 66929 – 66947.
- Schumann, R., Zhu, W., Feng, W., Fu, T. J., Riezler, S., & Wang, W. Y. (2024). “Velma: Verbalization embodiment of LLM agents for vision and language navigation in street view”. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 38, No. 17, pp. 18924-18933).
- Thinn, M., Thu, Y., Nwe, H., Yee, N., Myint, T., Thant, H. and Supnithi, T. (2020) “Machine translation of LaTeX based mathematical equations to spoken mathematics”, *24th International Computer Science and Engineering Conference (ICSEC)*. IEEE, 2020, 1–6.