

Latexify Math: Mathematical Formula Markup Revision to Assist Collaborative Editing in Math Q&A Sites

SUYU MA, Monash University, Australia
CHUNYANG CHEN*, Monash University, Australia
HOURIEH KHALAJZADEH, Monash University, Australia
JOHN GRUNDY, Monash University, Australia

Collaborative editing questions and answers plays an important role in quality control of Mathematics Stack Exchange which is a math Q&A Site. Our study of post edits in Mathematics Stack Exchange shows that there is a large number of math-related edits about latexifying formulas, revising LaTeX and converting the blurred math formula screenshots to LaTeX sequence. Despite its importance, manually editing one math-related post especially those with complex mathematical formulas is time-consuming and error-prone even for experienced users. To assist post owners and editors to do this editing, we have developed an edit-assistance tool, *MathLatexEdit* for formula latexification, LaTeX revision and screenshot transcription. We formulate this formula editing task as a translation problem, in which an original post is translated to a revised post. *MathLatexEdit* implements a deep learning based approach including two encoder-decoder models for textual and visual LaTeX edit recommendation with math-specific inference. The two models are trained on large-scale historical original-edited post pairs and synthesized screenshot-formula pairs. Our evaluation of *MathLatexEdit* not only demonstrates the accuracy of our model, but also the usefulness of *MathLatexEdit* in editing real-world posts which are accepted in Mathematics Stack Exchange.

CCS Concepts: • **Applied computing** → **Text editing**; • **Human-centered computing** → *Collaborative and social computing*.

Additional Key Words and Phrases: Q&A sites, Collaborative editing, Deep learning, Latex, Math

ACM Reference Format:

Suyu Ma, Chunyang Chen, Hourieh Khalajzadeh, and John Grundy. 2021. Latexify Math: Mathematical Formula Markup Revision to Assist Collaborative Editing in Math Q&A Sites. *Proc. ACM Hum.-Comput. Interact.* 5, CSCW2, Article 403 (October 2021), 24 pages. <https://doi.org/10.1145/3479547>

1 INTRODUCTION

As the foundation of science and engineering, mathematics has always been one of the most important subjects for students [32] among all levels of education. It can help reshape the students' reasoning, creativity, critical thinking and problem-solving abilities. On the other hand, mathematics is also one of the most challenging subjects for students [23, 32, 41]. Various mathematics community

*Corresponding author.

Authors' addresses: Suyu Ma, Monash University, Faculty of Information Technology, Melbourne, Australia, suyu.ma1@monash.edu; Chunyang Chen, Monash University, Faculty of Information Technology, Melbourne, Australia, chunyang.chen@monash.edu; Hourieh Khalajzadeh, Monash University, Faculty of Information Technology, Melbourne, Australia, hourieh.khalajzadeh@monash.edu; John Grundy, Monash University, Faculty of Information Technology, Melbourne, Australia, john.grundy@monash.edu.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2021 Association for Computing Machinery.

2573-0142/2021/10-ART403 \$15.00

<https://doi.org/10.1145/3479547>

Question and Answer (Q&A) sites have been developed. They provide a platform for students to ask about anything related to the mathematics with fast feedback from peers and experts. Mathematics Stack Exchange [7] is an example of such a Q&A site for people studying mathematics at any level and for professionals in related fields. Since its launch in 2010, it contains 1.3 million questions and 1.7 million answers from 661 thousand users [6].

However, the dramatic growth of posts and users on such Q&A sites poses a severe challenge to the quality assurance for site content. For example, some novice users may post questions and answers with grammar errors, misuse of abbreviations, blurred screenshots of formulas, and the lack of important information (e.g., context or referenced resources) for understanding questions and answers. Such quality deterioration can negatively influence the readability and understandability of posts, and may further discourage the participation of users [37]. To avoid the quality decay of the sites, Q&A sites like Mathematics Stack Exchange provide official recommendations for effective question writing [3] and answer writing [2].

Although all users are encouraged to follow the quality assurance policies, many users may still violate them carelessly or unintentionally, and some users may not even be aware of the existence of such policies. To ensure the site quality [40], Mathematics Stack Exchange encourages users, especially experienced users [8], to collaboratively edit the posts to make them comply with the site quality standards. According to our analysis in Section 3, among 2,886,174 posts including questions and answers (as of Mar 2020), 1,318,753 (45.69%) of them have been edited at least once. Post edits involve not only minor corrections of misspellings and grammar errors, but also math-related such as formatting math formula for better readability, fixing mathematical mistakes.

Although collaborative editing is beneficial for the community [36], there are still three problems with such mechanism. First, it requires significant community effort, especially from high-reputation users to edit the posts directly and/or approve the edits by the other users. Second, some errors in the original posts, especially relatively complicated ones such as math formula errors or format are difficult to spot, as they may require a good understanding of the question or answer content. Third, all these collaborative edits are reactive to existing errors which may have already harmed the readers of the posts before edits, or made it difficult for those who want to help to answer the questions.

Therefore, in addition to collaborative editing for reactive quality assurance, we also need a more proactive mechanism of quality assurance which could check a post before it is posted, spot the potential issues in the post, and remind the post owner to fix issues if any. Towards that target, there are some research works investigating the automated revision of Q&A posts [13, 17, 36, 51]. However, different from other Q&A sites (e.g., Quora¹ and Stack Overflow²), there are some domain-specific edits required on Mathematics Stack Exchange due to the mathematics characteristics which cannot be addressed by their automated methods. By analysing the topics in the comments of post edits and empirical study of historical math-related edits in Section 3, three editing types emerge including formula latexification, LaTeX formula revision, formula screenshot transcription, involving 516,010 related edits. Considering the wide range of post content and formats involved in post edits (see Table 2 for examples), it would require significant manual effort to develop a complete set of rules for representing editing patterns.

This challenge motivates us to develop a data-driven deep-learning based janitor for recommending post owners or other users potential math-related edits, by automatically learning from historical edits. Note that Mathematics Stack Exchange encourages users to use LaTeX, a computer programming language used in typesetting technical documents [4], for writing mathematical

¹<https://www.quora.com/>

²<https://stackoverflow.com/>

formulas. It can be rendered to clear and nice-looking formulas by Mathjax [12]. In this work, we formulate the post editing recommendation to a translation task, in which an original post is “translated” into the edited post. This includes translating erroneous/unformatted math formula to the correct/formatted one (textual LaTeX edit) and math formula screenshots to more readable ones in LaTeX (visual Latex edit). We design two models for these two tasks and define them as textual edit model and visual edit model. We adopt transformer-based model for textual edits and use DenseNet [31] and Long Short-term Memory (LSTM) [30] models for visual edits. To improve the performance of our tool, we adopt the sentence normalization which shortens the length of the input by directly copying the non-math content when fixing formula error and formula formatting (Section 4.3). We also improve the basic screenshot transcription method by adding image similarity into the inference phase for boosting the quality of generated LaTeX. We name our tool as Mathematical Latex Editor (MathLatexEdit).

The trained `MathLatexEdit` automates the math-specific edits, thus removing the need for prior manual editing rule development. Our work is the first one to explore how to support collaborative editing in math-related question and answer sites. Our work fills in the gap between math-related post editing and cooperative work. This tool will not only help post owners reduce minor math-related issues before posting their questions and answers, but also help post editors improve their editing efficiency. Furthermore, the identified issues together with the recommended corrections will help novice post editors learn community-adopted editing patterns.

To train and test our model, we develop a text differencing algorithm to collect a large dataset of original-edited sentence pairs from post edits in Mathematics Stack Exchange for textual edit model and prepare a large dataset of LaTeX formula and image pairs for visual edit model. Our results show that our approach outperforms other rule-based or deep learning based baselines for post edit recommendation. Apart from the model performance, we also conducted a user study to confirm that participants with our tool can finish the editing with less time but higher precision, recall and satisfactoriness. The field study also shows its usefulness in which the first author acts as a novice post editor that has little post editing experience. Based on the post editing recommendations by our tool, he edited 80 posts and submit the edits to Mathematics Stack Exchange, and 78 of them were accepted. That is, for each accepted post edit, at least two trusted contributors considered that the edit had significantly improved the post quality.

We make the following key contributions in this paper:

- We conducted an empirical study of collaborative post editing in Mathematics Stack Exchange including the editing types and editing content, identifying the need for an edit assistance tool for mathematics formulae.
- We developed `MathLatexEdit`, a deep learning based edit assistance tool that can latexify math expression, revise LaTeX formulas, convert formula screenshots in Mathematics Stack Exchange to LaTeX sequence for further MathJax rendering to improve the post quality and readability. Although the current work studies Mathematics Stack Exchange, our data analysis method and deep learning approach could be applied to other social systems which is discussed in Section 7.1.
- We evaluated `MathLatexEdit` from two perspectives, including the quality of post edit recommendation based on a large-scale dataset and the usefulness to assist a novice post editor in editing unfamiliar new posts.

2 RELATED WORK

2.1 Mathematics Q&A Sites

Online question and answer (Q&A) sites are platforms for participants to ask and answer questions. With the power of crowd-sourced answers, Q&A sites, such as Quora³ and Stack Overflow⁴, are more and more popular with accumulating millions of questions and answers in different domains [11, 18, 20, 39]. Due to the importance and challenges of learning math, math-specific Q&A sites have also been launched, such as Mathematics Stack Exchange⁵ for students and MathOverflow⁶ for professional and expert researchers. MathOverflow has attracted great attention from researchers. Montoya et al. [44] model MathOverflow as a social network for analyzing its social achievement and centrality. Tausczik et al. [49] investigate the collaboration patterns when solving a research-level mathematical problem in MathOverflow.

However, few researchers explore the Mathematics Stack Exchange. As one of the most popular mathematics Q&A sites with 635 new questions and 200K visits per day [6], Mathematics Stack Exchange deserves more attention from our research community. To enhance the post quality in Mathematics Stack Exchange, our study is the first work to explore its collaborative editing patterns and propose a deep learning method to assist post owners and editors to revise low-quality posts.

2.2 Collaborative Editing in Q&A Sites

Although collaborative editing is widely studied in some user-generated content (UGC) communities (e.g., Wikipedia [5], wikiHow [1]), there are relatively few works focusing on Q&A sites. Collaborative editing can help improve the quality control in Q&A sites by converting low-quality posts to higher-quality ones [40]. Li et al. [37] demonstrate that it will not hurt the user engagement, despite the quality improvement. Ford et al. [26] show that collaborative editing with mentors can further improve engagement in Q&A site. Choi and Yla [21] find that the moderators in collaborative editing can help resolve the conflict of using tags to describe the questions which is caused by users' background and understanding differences.

Although collaborative editing is important for the site quality, it takes much human effort to realise. To assist collaborative editing in online communities, some machine learning based methods have been proposed. Li et al. [36] and Chen et al. [13] developed a classifier to automatically predict if a post needs edits or what type of edits are required e.g. adding links, images, updating format, etc. Chen et al. [17] proposed a deep learning based model to automate some minor revisions such as misspelling, grammar errors and keyword formatting in sentences of the post.

Our MathLatexEdit tool differs in two key aspects. First, most related works above focus on the most popular programming Q&A site, Stack Overflow [17, 37, 40, 51], while we focus on the Mathematics Stack Exchange. We carried out an empirical study to explore the domain-specific edits in this math-related Q&A site. Second, in addition to their models based on the text of posts, we are also working on a more challenging type of collaborative editing, i.e., converting formula screenshots to LaTeX sequence that bridges the gap between visual and textual information.

2.3 Grammar Error Correction

We formulate the post editing recommendation in this work as a translation task i.e., translating the text, buggy latex commands and formula screenshots to corresponding correct latex commands. There are many research works on automated textual grammar error correction with machine

³<https://www.quora.com/>

⁴<https://stackoverflow.com/>

⁵<https://math.stackexchange.com/>

⁶<https://mathoverflow.net/>

learning [33, 43, 57] i.e., detecting and fixing grammar error of the original natural-language text. For example, Junczys-Dowmunt and Grundkiewicz design an approach that can automatically correct grammar errors with phrase-based Statistical Machine Translation (SMT) method [33]. Mizumoto and Matsumoto [43] and Yuan et al. [57] recommend ranked grammar error correction with SMT method and a ranking method. Unlike traditional SMT methods, Neural Machine Translation (NMT), such as RNN-based methods and transformer-based methods [52], utilize sentence context information and joint all the components in the training process. CNN based Seq2Seq and quality estimation methods are used by Chollampatt and Ng [22] to automatically estimate the quality of GEC sentences. Grundkiewicz and Junczys-Dowmunt combine SMT and neural machine translation to automated Grammatical Error Correction [29]. Different from their works, we are the first to target at mathematic-specific revisions i.e., the content change of latex commands by taking the mathematic characteristic into the consideration.

2.4 Mathematics Formula Accessibility

It is crucial for the community to make mathematical formula accessible [23, 41], as many users just take a screenshot of mathematics formula and put it online which can significantly reduce the readability of normal users and engagement of users with vision impairment. To assist the conversion of mathematics formula screenshots to relevant representation in LaTeX, researchers propose many different algorithms. A commercial OCR tool is used by Garain et al. [28] to classify text, and the unrecognized patterns were further analyzed to detect mathematics formulas. Based on larger symbols and blank spaces, T waaliyondo el al. [50] divide the formulas into sub expressions and represent them as a tree. Suziki et al. [28] use a similar approach with a minimum cost spanning-tree algorithm. The commercial software InftyReader [48] is based on this work. Inspired by the image captioning tasks [14, 19, 55], Deng et al. [24] designed a more advanced deep learning based model.

Compared with these general works, our image transcribing algorithm is the first targeting at collaborative editing in the real Mathematics Stack Exchange. Our approach not only takes the textual Latex edit into consideration and is also more advanced by incorporating DenseNet and inference with additional visual similarity for converting screenshot into Latex representation described in Section 4. Different from these theoretical works only focusing on the model performance on testing dataset, we also carried out a user study and a field study in Section 6 to verify the usefulness in assisting with real-world post edits.

3 COLLABORATIVE EDITING ANALYSIS OF MATHEMATICS STACK EXCHANGE

Different from other Q&A sites (e.g., Quora and Stack Overflow), there are some domain-specific edits required on Mathematics Stack Exchange due to the mathematics characteristics which cannot be addressed by automated methods. We downloaded the latest data dump⁷ of Mathematics Stack Exchange which contains 2,886,174 posts (including 1,216,368 questions and 1,669,806 answers) and all post edits since its launch on July 20, 2010 to March 1, 2020. Based on this large dataset, we carried out an empirical study of post edits in Mathematics Stack Exchange to understand the characteristics of post editing in Mathematics Stack Exchange and to motivate the required tool support. This empirical study allows us to re-frame our research from the perspectives of quality control and beneficial aspects of our tool.

⁷<https://archive.org/download/stackexchange>

Table 1. Topics of post-edit comments

ID	Topic Name	Keywords
1	spelling & grammar	grammatical error, spelling, typo
2	adding links	wikipedia link, broken link, linked
3	explaining the question	descriptive, explanation, definition
4	converting image	replace image, latexifying image, transcribed image
5	revising mathematical content	algebraic, mathematical, mathematics
6	changing format	adjust format, formatting, formatted
7	improving readability	improve readability, easier read, make readable
8	modifying LaTeX formula	tex, improving latex, tex improvement, applied mathjax

3.1 What are the edits about?

In Mathematics Stack Exchange, there are three kinds of post information which can be edited – question tags [15, 16], question title, and post (question and answer) body [40]. Question-title and post-body editing are of the same nature (i.e., sentence editing), while question-tags editing is to add and/or remove the set of tags of a question.

As of March 1, 2020, there have been in total 2,696,115 post edits. Among them, 334,136 (12.39%) are question-title edits, 286,715 (10.63%) are question-tag edits, and the majority of post edits (2,075,264 (76.97%)) are post-body edits. The tags of 268,620 (22.08%) questions, the titles of 243,184 (20.00%) questions, and the body of 1,075,420 (37.26%) posts have been edited at least once. 63.21% of these edits are self-edits by the post owners and the other 36.79% are collaborative edits by other post editors. Overall, post-body edits make up the majority of post edits, and post-body editing are more complex compared with revising title or question tags. Therefore, we focus on post-body edits in this work. Hereafter, post edits refer to post-body edits, unless otherwise stated.

3.2 Who edited posts?

Among all 2,075,264 post edits, 1,228,763 (59.21%) are self-edits by the post owners, 846,501 (40.79%) are edits by other users. This data suggests that an edit assistance tool may benefit the Mathematics Stack Exchange community from three perspectives.

First, the tool can highlight issues in posts that post owners are creating and proactively remind them to fix the issues. This reduces the need for editing after the creation and ensures the post quality in the first place. Second, an edit assistance tool that recommends edits can improve the efficiency in editing others' posts by providing a reliable starting point. Third, the post editors can use our tool to learn to correct minor domain-specific issues. Such small editing tasks can provide a mechanism of legitimate peripheral participation [34] for novice users who have no experience in Mathematics Stack Exchange. This may help to on-board novice users in Mathematics Stack Exchange and improve the quality of their post edits.

3.3 What has been edited?

To understand what post edits are about, we analyzed the comments attached to the post edits. In Mathematics Stack Exchange, when users finish editing a post, they can add a brief comment to explain the editing reasons. We collected all post-edit comments and applied standard text processing steps to post-edit comments such as removing punctuations, lower-casing all characters, and excluding stop words.

We extracted the topics from users' editing comments to understand their editing content. To extract common editing types, we adopted the Latent Dirichlet Allocation (LDA) [10] model to

analyze the post-edit comments. LDA is a statistical model for discovering abstract topics that occur in a collection of documents in which each topic consists of a set of keywords. A significant limitation of LDA is that it considers only single words (i.e., unigrams). However, a single word may not capture the exact semantics of the post-edit comments. In contrast, phrases that are composed of several words are more intuitive to understand the intention behind post edits, such as “latexifying image” instead of “image”, “improving latex” rather than “latex”. Therefore, these multi-word phrases must be recognized and treated as a whole in LDA model.

We adopted a simple data-driven and memory-efficient approach [42] to detect multi-word phrases in post-edit comments. In this approach, phrases are formed iteratively based on the unigram and bigram counts, using the following formula $score(w_i, w_{i+1}) = \frac{count(w_i, w_{i+1}) - \theta}{count(w_i) \times count(w_{i+1})} \times N$. The w_i and w_{i+1} are two consecutive words. θ is a discounting coefficient to prevent infrequent bigrams to be formed. That is, the two consecutive words will not form a bigram phrase if they appear as a phrase less than θ times in the corpus. N is the vocabulary size of the corpus. In this work, we experimentally set θ as 10 and the threshold for score as 10 to achieve a good balance between the coverage and accuracy of the detected multi-words phrases.

Our method can find bigram phrases that appear frequently enough in post-edit comments compared with the frequency of each unigram, such as “improve readability”. However, bigram phrases like “it is” will not be formed because each unigram also appears very frequently in the text. All these phrases are then concatenated with underline like “improve_readability” in the corpus of post-edit comments, and then we use the LDA model to extract the topics.

We extracted 8 topics with corresponding keywords shown in Table 1. Note that we annotated the topics name with our own summarization based on the topic keywords. Different from individual frequent keywords, these topics provide finer-grained information. Apart from common types mentioned above, there are also some domain-specific editing patterns related to mathematics formula (#5 revising mathematical content, #8 modifying LaTeX formula) and readability (#4 converting image, #6 changing format, #7 improving readability). These two common editing types represent the community norms that Mathematics Stack Exchange commits its effort to maintain.

3.4 What efforts have been committed to the domain-specific edits?

As mentioned in Section 3.3, many post edits are domain-specific and highly related to mathematics, especially mathematics formula and readability, according to the analysis of the editing comments. To further explore what edits are about, we extracted the edits with editing comments including domain-specific words such as “formula”, “LaTeX”, “math”, etc. By traversing the commented editing history in Mathematics Stack Exchange, we collected 155,369⁸ domain-specific edits in total. We then randomly selected 100 of them for manual inspection. According to our observation, there are three common mathematics domain-specific editing patterns, as seen in Table 2.

First, 29 edits are to change a plain formula into LaTeX format with MathJax rendering for a clearer view, resulting in better readability. It may be the first time for some users to join the Mathematics Stack Exchange, so they write a plain math formula in the post. But editing it into LaTeX can distinguish the mathematics formula from the plain text so that other users can easily understand the meaning of the post, leading to higher possibility of responses. For instance, “x - root2” is converted into “ $x - \sqrt{2}$ ” in the first example of Table 2. Compared with natural English words, LaTeX mathematical expression highlights these numerical tokens and help readers easily capture the key point of the posts.

Second, 35 edits focus on correcting mistakes in formulas in order to clarify the content. Some tokens in formulas are missed or misspelled. Even the experienced users may make mistakes when

⁸Note that only 726,342 edits contain comments, so the number is highly underestimated.

Table 2. Types of domain-specific accessible edits

<p>1 changing formula format</p> <p>I know that between any two real numbers there exists an irrational number. Proof: Let $x < y$ be two real numbers. Then $x - \sqrt{2}$ and $y - \sqrt{2}$ are also real numbers. There is a rational number r such that $x - \sqrt{2} < r < y - \sqrt{2}$. Adding $\sqrt{2}$ to both sides of these equations, and we have $r + \sqrt{2}$. We know that r is rational and $\sqrt{2}$</p>	
<p>2 correcting mistakes in formula</p> <p>To establish the hint, just pick some $a \in A$ such that $g(a) = 1$, and show that $\langle a \rangle \cong \mathbb{Z}$, $\langle a \rangle \cap \ker(g) = \{0\}$, and $A = \ker(g) \oplus \langle a \rangle$. $\text{mathrmker}(g) = \{0\}$</p>	<p>To establish the hint, just pick some $a \in A$ such that $g(a) = 1$, and show that $\langle a \rangle \cong \mathbb{Z}$, $\langle a \rangle \cap \ker(g) = \{0\}$, and $A = \ker(g) \oplus \langle a \rangle$.</p>
<p>3 replacing image with LaTeX formula</p> <p>Take a $y \in [c, d]$. From the Lagrange theorem it follows that $\exists \theta \in (0, 1)$ so that:</p> <div style="border: 1px solid red; padding: 5px; margin: 5px 0;"> $\frac{I(y+k) - I(y)}{k} = \int_a^b \frac{f(x, y+k) - f(x, y)}{k} dx$ $= \int_a^b \frac{\partial f}{\partial y}(x, y + \theta k) dx$ </div> <div style="background-color: #e0f0e0; padding: 5px; margin: 5px 0;"> $\frac{I(y+k) - I(y)}{k} = \int_a^b \frac{f(x, y+k) - f(x, y)}{k} dx$ $= \int_a^b \frac{\partial f}{\partial y}(x, y + \theta k) dx$ </div>	

writing math formulas, especially for the complicated ones. Since the math formula may be the most important content, many edits are targeting at it for more accurate information. For instance, extra \backslash is removed in second example of Table 2, which improves the quality of the post.

Third, 8 edits are to convert the blurred formula images into LaTeX which is further rendered to a vector graph so that users can zoom in or out for better view in the third example of Table 2. There are mainly two kinds of blurred formula images, which includes screenshots and pictures captured by cameras. Compared with the blurred formula images, the LaTeX format also makes it easy to be indexed by the search engine, resulting in the searchable text.

The rest edits are not related to mathematics, such as adding or deleting the content in posts, correcting the grammar errors in posts, or changing the display formats.

3.5 How many mathematic-specific edits are there?

In Mathematics Stack Exchange, all mathematical formulas should be written in LaTeX format with further MathJax rendering for clear visualization. For each of the three types of mathematics-specific edits (formula latexification, LaTeX revision, and screenshot transcription) appear most often in the last section, we observe the corresponding detailed text changes for detecting instances of a particular type of edits and their frequencies as follows:

- To render better readability of mathematic formula, Mathematics Stack Exchange encourages users to annotate their formula in LaTeX with \$ at the beginning and end (Table 2.(1)). Users also need to use the LaTeX grammar like some special symbols. The revision from plain math formula in text into LaTeX is called as *formula latexification*.
- To improve the accuracy of formulas, users are encouraged to revise others' formula LaTeX in posts which is called *latex revision* (Table 2.(2)) in this work.
- To improve the readability and accessibility of math formula, the embedded image/screenshots which require a special link (Table 2.(3)) ending with postfix including .png, .jpg, .gif, .bmp and .tiff need to be converted to formula in LaTeX with \$ at the beginning and end. We refer to this kind of revision of screenshot links as *screenshot transcription*.

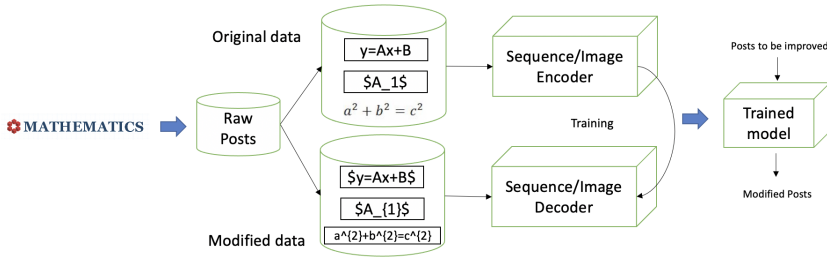


Fig. 1. Workflow of our approach

By differencing the original post body and the edited post body and following last three patterns, we count the number of different types of edits. There exist 627,251 math-specific edits in total, in which 516,010 (24.86% out of all post body edits) edits are following three types.

- 169,006 (169,006 / 516,010 = 32.75%) post edits include *formula latexification*.
- 288,947 (55.99%) post edits include *LaTeX revision*, and 548,230 formulas are revised in total.
- For *screenshot transcription*, 59,141 (11.46%) post-body edits include that conversion.

Considering the diversity of text, formula, pixels and context involved in math-related edits, it would require significant manual effort to develop and validate a complete set of rules for representing editing patterns. For example, “root” should be converted to “sqrt” in latex, but will not be changed in many other contexts. So, it is impractical to enumerate all such cases. An advanced approach is highly needed for automated editing.

Summary: 45.69% posts in Mathematics Stack Exchange have been edited involving a variety of editing types, including fixing grammatical errors, clarifying the meaning of a post, formatting the post and adding related resources or hyperlinks. In addition, there are also many domain-specific edits like formula latexification, LaTeX revision, screenshot transcription for better readability and accessibility. These 516,010 math-specific edits (24.86% out of all post body edits) require much more human effort to maintain in the community due to the complexity of math formulas. Therefore, a post edit recommendation algorithm is needed to assist users in Mathematics Stack Exchange to correct math-specific errors and assist these three math-related edits for ensuring the post quality.

4 RECOMMENDING LATEX EDITS BY DEEP NEURAL NETWORK

The three types of post edits related to math formula in our empirical study highlight the community efforts for ensuring the post quality in Mathematics Stack Exchange. Unfortunately, these efforts and revisions are implicit knowledge in millions of post edits. Considering the diversity of post editing types and contexts, it would require significant human effort to build a complete set of rules to deal in all different situations. Therefore, we developed a deep-learning based approach which can automatically improve the post editing patterns from historical post edits, and recommend edits to the new posts based on the learned editing knowledge.

4.1 Overview of our MathLatexEdit Approach

The workflow of our approach is shown in Fig 1. Given three types of math-specific edit, we separate them into two tasks i.e., textual LaTeX edit (*formula latexification*, *LaTeX revision*) and visual LaTeX edit (*screenshot transcription*). Our approach first collected a large corpus of original-edited sentence pairs of modifying math formulas for subsequent textual LaTeX edit and synthesized a large corpus of image-formula pairs for model training for subsequent visual LaTeX edit (Section 4.2). For

textual LaTeX edits, our approach trained a transformer based model on a large parallel corpus of original-edited sentence pairs (Section 4.3). For visual LaTeX edits, our approach adopted an encoder-decoder model for converting the formula screenshot to LaTeX representation based on synthesized image-formula pairs (Section 4.4). We specify the implementation details in Section 4.5.

4.2 Data collection

4.2.1 For formula latexification and LaTeX revision tasks. A post may have been edited several times. Assume a post has N versions, i.e., undergoing $N - 1$ post edits. For each post edit i ($1 \leq i \leq N - 1$), we collect a pair of the original and edited content. The original content is from the version i of the post before the edit, and the edited content is from the version $i + 1$ of the post after the edit. Then, we split the content into sentences.

We then align the sentence list $oList$ from the original content and the sentence list $eList$ from the edited content. For a sentence in the $oList$, if the similarity score of the most similar sentence in the $eList$ is above a threshold, the two sentences are aligned as a pair of original-edited sentences. To calculate the similarity between one original sentence l_o and edited sentence l_e , we calculate the Levenshtein distance [35] $dis(l_o, l_e)$ between two sentences. The similarity threshold should be set to achieve a balanced precision and recall for sentence alignment. Therefore, we experimentally set the threshold at 0.9 in this work. Note that some small edits may not influence readability a lot, however, the aggregating effect of several small issues in one post are not human-tolerable. Therefore, we still take them into consideration during data collection.

We also filter out non-mathematical and noisy edits using the rule below: We remove the sentence pairs that do not include LaTeX formulas in original and edited sentences and sentence pairs in which the LaTeX formula does not change. We also remove too long (char number > 256) or too short (char number < 10). In total, we collect 220,093 sentence pairs before March 1, 2020.

4.2.2 For formula screenshot transcription task. In Mathematics Stack Exchange, most users follow the guidelines by posting mathematics equations into LaTeX rather than as an image. To collect the data for training our visual LaTeX edit model, we collected all LaTeX mathematics equations from Mathematics Stack Exchange and then rendered them to an image with automated scripts. First, we extract mathematics formulas using regular expressions i.e., extracting text within special annotations like “`begin\{equation\}(.*)end\{equation\}`” and “`\$([\^\$]*)\$`”. By matching the raw posts content with regular expressions, we collected 5,505,098 raw LaTeX snippets.

Second, to filter out some noisy data, we removed the sequences without any mathematics characters such as “+”, “`\frac`”. We also removed duplicate LaTeX sequences and too long (char number > 256) or too short (char number < 10) LaTeX sequences.

Third, we tokenized the 2,068,744 remaining LaTeX sequences into separate words, especially the domain-specific tokens such as “`\sum`”, “`\frac`” and “`\sin`” as one word. To avoid ambiguity i.e., the same image can be generated by different LaTeX, we developed a LaTeX parser to keep the same LaTeX formatting. For instance, we replace “`A^{\{c\}}_{\{2\}}`” with “`A_{\{2\}}^{\{c\}}`”, change “`\over`” to “`\frac{\{c\}}{\{2\}}`” and delete “`\label{\}`”. We converted these LaTeX formulas to images with *pdfLaTeX* and excluded any formulas that failed to compile. Note that these synthesized images are different from the real-world screenshots. The synthesized images are generated following the same rules while real-world screenshots differ greatly from users to users. To bridge the gap between the GUI screenshots from the two resources, we applied an image augmentation method to transform the synthesized images to mimic those found in real-world images. To do this, we randomly applied different sizes and resolution (i.e., DPI(dots per inch)) when rendering the synthesized images based on the collected LaTeX formula. Some resulting formula screenshot examples by the image augmentation can be seen in Fig 2.

$$\{Y_n\}_{n=1}^N \quad \oint_L f(z) dz = 0 \quad M_2(\mathbb{R}) / \{0_{M_2(\mathbb{R})}\} \quad \vec{n}_1 = k \vec{n}_2$$

Fig. 2. Examples of synthesized data

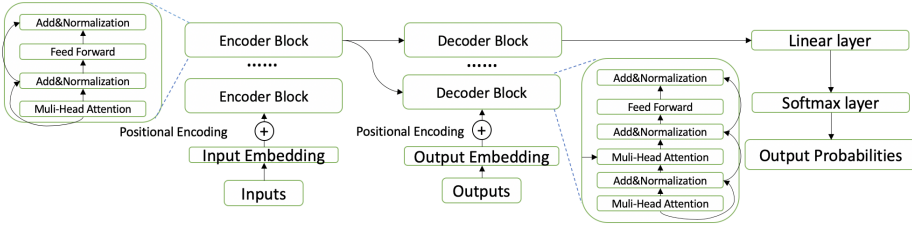


Fig. 3. Textual LaTeX edit structure of MathLatexEdit

4.3 Textual Latex Edit Recommendation

The textual LaTeX edit recommendation can be treated as a machine translation problem by treating the original post sentence as input and edited sentence as output. Therefore, we adopt neural machine translation model [27, 54] to learn the mapping from the source sentence to the target sentence.

As shown in fig 3, an attention-based transformer architecture is used for formula latexification and LaTeX revision. Given the source word tokens (x_1, \dots, x_N) , the goal is to predict the target word tokens (y_1, \dots, y_T) . The source word tokens are the original post sentence, and the predicted output word tokens are the edited post sentence.

The performance of deep learning models heavily depends on the quality of the training data. In particular, our transformer model is sensitive to the input length i.e., the longer input sequence always results in worse performance as it is hard for the model to capture the long semantic especially for those long complex math formulas. To mitigate that issue, we develop a normalization way to preprocess the input before feeding it to the model. Many sentences within our dataset is of both mathematical formula and natural-language part. According to our observation, the non-math related content is rarely changed and math related content is always of special symbols like punctuations (e.g., "\$+*=*"), numbers (e.g., 2, 3.14), variables (e.g., a, b, x, y) or commands in LaTeX (e.g., log, sin). Therefore, we manually construct a list of rules for detecting the potential mathematical content within the sentence, and replace the non-math part with the same placeholder symbol in the input. To further shorten the length of input sequence, we also normalize the number with another placeholder symbol since it is rarely changed during the edit. For example, the original sentence "my first though was to factor by doing $(2 + e^x - e^{-x}) / (e^{-x} + 1)$ but that negative in the denominator is not letting me solve the problem ?" is preprocessed into "COMMON_WORDS $(2 + e^x - e^{-x}) / (e^{-x} + 1)$ COMMON_WORDS", which highly shortens the length of the input.

The normalized sentence pairs are used to train or test our transformer model. Given an original sentence to be edited, the trained model will change it into an edited sentence. The special symbols are then mapped back to the original domain-specific words in a post-processing step. For instance, the edited sentence "COMMON_WORDS $\frac{2 + e^x - e^{-x}}{e^{-x} + 1}$ COMMON_WORDS" is

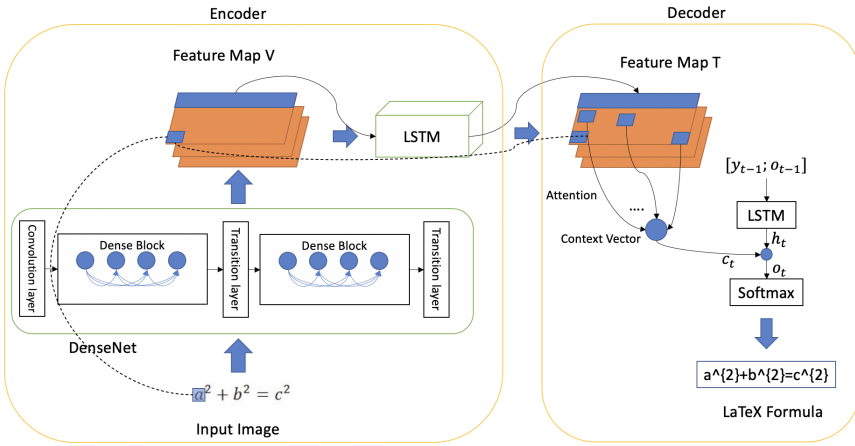


Fig. 4. Visual LaTeX edit structure of MathLatexEdit

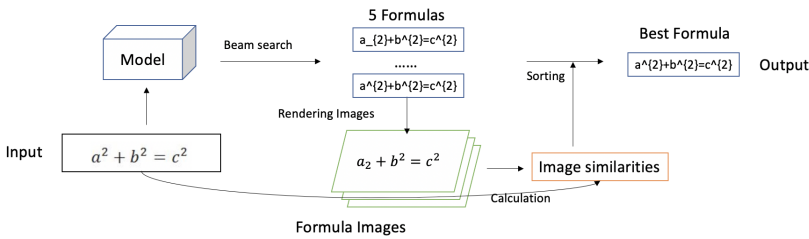


Fig. 5. Overview of inference with image similarity

mapped back to "my first thought was to factor by doing $\frac{2 + e^{-x} - e^{-x}}{e^{-x} + 1}$ but that negative in the denominator is not letting me solve the problem?".

4.4 Visual Latex Edit Recommendation

Our MathLatexEdit tool applies an encoder-decoder structure for screenshot transcription, in which the encoder uses DenseNet, and the decoder uses LSTM with attention mechanism. The overall structure is shown in Fig 4.

To get the feature maps of the input images, DenseNet is first used in the encoder to extract a feature map V . Unlike traditional Convolutional Neural Network (CNN), DenseNet connects each layer to every subsequent layer [31, 53]. The output features of DenseNet contain sequential order information. Thus we use another LSTM encoder to re-encode each row of DenseNet’s output feature map. Based on the feature map T , we use LSTM and an attention mechanism [52] as decoder to generate a sequence of predicted LaTeX tokens.

After training the visual LaTeX edit model, we use it to generate the LaTeX sequence l for a mathematics screenshot s . As seen in Fig 5, we first adopt a beam search approach to select the top-5 candidates. For each candidate, we render it into a formula image to compare the similarity of it with the original one and then select the most similar one as the generated result.

Given the input screenshot, the generated LaTeX sequence should have the maximum log probability $p(l|s)$. However, generating a global optimal LaTeX sequence has an immense search

space. Therefore, we adopt a beam search to expand only a limited set of the most promising nodes in the search space.

After selecting the top-5 candidates with the highest probability, we further render them into mathematical formula images. We compare these rendered images with the original input screenshot, and re-rank 5 candidates by image similarity. To calculate image similarity, we firstly crop and resize these two images to same size. We then convert images to a binary image with pixel values as either 0 or 1, where 0 as black and 1 as white. For each column, we can convert a sequence of 0 and 1 into a value with binary system. Let a_i be the image array that is converted from original image i . Let a_j be the image array that is converted from rendered image j . We calculate the Levenshtein distance between a_i and a_j as $dis(a_i, a_j)$. We can get the image similarity as $1 - dis(a_i, a_j) / \max(len(a_i), len(a_j))$, where $\max(len(a_i), len(a_j))$ is the max length of a_i and a_j . Based on the image similarity of 5 candidates, we select the one with the highest similarity and its corresponding LaTeX sequence as the generated result of MathLatexEdit.

4.5 Implementation

MathLatexEdit is based on Pytorch⁹ (textual LaTeX edit recommendation model) and Torch¹⁰ (visual LaTeX edit recommendation model) and all experiments were run on a 11GB NVidia TITAN Xp. The following settings are same for the two models. Minibatch stochastic gradient descent is used to optimize the parameters. The initial learning rate is set as 0.1. The training epoch is set as 25 and perplexity is used to select the best model during the validation step. Once the validation score does not decrease, we halve the learning rate. Beam search is used during the test step, and the beam size is 5. We also release the source code¹¹ of this project.

5 EVALUATING THE QUALITY OF MATHLATEXEDIT

Our MathLatexEdit tool aims to help Mathematics Stack Exchange Q&A site post owners and editors to more effectively and collaboratively edit the math-related content in the post. The quality of the generated recommendation will hence greatly affect the utilization of MathLatexEdit by the community.

5.1 Dataset

For textual LaTeX edit recommendation, from 8,530,558 posts in Math Stack Exchange, we collected 219,420 original-edited sentence pairs about formula latexification and LaTeX revision. We randomly took 186,507 (85%) of these sentence pairs as the training data, 10,971 (5%) as the validation data and 21,942 (10%) as the testing data to evaluate the quality of recommended edits by our tool.

For visual LaTeX edit recommendation, we collected 2,068,744 LaTeX sequences. Based on these extracted formulas, we generated 1,000,000 image-formula pairs. We randomly selected 900,000 (90%) of these pairs as the training data, 50,000 (5%) as the validation data to tune model hyperparameters, and 50,000 (5%)¹² as the testing data ($test_{synthesize}$) to evaluate the quality of converted LaTeX sequences by MathLatexEdit. Apart from these synthesized data, we also collected the real-world post editing history where human editors have converted a posted math formula screenshot to LaTeX. For each post edit, we compared the original and edited posts to check if some images in the original post are replaced by LaTeX sequence in the corresponding position of the edited post. From 12,463 image-formula pairs that the images are manually converted to LaTeX sequences by editors,

⁹<https://pytorch.org>

¹⁰<http://torch.ch/>

¹¹[inhttps://github.com/astra1230/MathFormGen](https://github.com/astra1230/MathFormGen)

¹²We only use 5% for testing as the image processing is much slower than text processing.

we randomly selected 100 of them as another testing set ($test_{real}$) to evaluate the performance of MathLatexEdit.

5.2 Baselines

Apart from our own MathLatexEdit, we selected other four methods as baselines for our comparisons. For textual LaTeX edit recommendation, the first baseline is a grammar error correction sequence to sequence (Seq2seq) model that contains a bidirectional RNN as an encoder and an attention-based decoder [56]. The other baseline is the phrase-based Statistical Machine Translation (SMT) model [46] specifically designed for sentence correction. To check the influence of sentence normalization, we also take the derivative of our approach as the baseline i.e., our own MathLatexEdit without sentence normalization. For all deep-learning models, we use the same training data to train the model.

For visual LaTeX edit recommendation, the first baseline is the InftyReader, a commercial software mathematical expression recognition system. This tool is an OCR-based system, which combines symbol recognition and structural analysis phases [48]. The second baseline is the deep learning based model, WYGIWYS, which is specially designed to convert formula images to LaTeX sequences [24]. We also add our approach derivative i.e., MathLatexEdit without image similarity as another baseline.

5.3 Evaluation metrics

We adopt the BLEU (BiLingual Evaluation Understudy) [47] for evaluating the quality of textual and visual LaTeX edit recommendation. BLEU is an automatic evaluation metric widely used in machine translation studies. It calculates the similarity of machine-generated translations and human-created reference translations (i.e., ground truth).

GLEU (Generalized Language Understanding Evaluation) is also used for evaluating textual LaTeX edit recommendation. In the GEC field, recent released shared tasks have prompted the development of GLEU for evaluating GEC approaches [45]. GLEU is a customized metric from the BLEU score which is widely used to evaluate the machine-translation quality. [56] It is independent of manual-annotation scheme and requires only reference sentences (without annotations of gold-standard edits). Recent study shows that GLEU has the correlation with human judgments of GEC quality and effort [45]. Since it requires the input sequence, we do not use it for evaluating visual latex recommendation.

Apart from BLEU score, we also adopt the image similarity to check the quality of visual LaTeX edit recommendation by measuring the image similarity between the given screenshot and rendering image based on the generated Latex. As the generated LaTeX can also be used to render the math image, we use the predicted LaTeX sequences and ground-truth LaTeX sequences to generate pair of images with the same resolution. We binarize the pair of images so that the pixel values are all 0 and 1, where 0 means black and 1 means white. Then we convert the image into a one dimension array and remove the elements that only contain 1 value. The following steps for image similarity is the same as what we use in Section 4.4.

5.4 Evaluation Results

We report the evaluation results of MathLatexEdit from two aspects i.e., the performance of MathLatexEdit in recommending edits to formula latexification and LaTeX revision, and how accurate MathLatexEdit can be to generate the LaTeX sequence for a given math screenshot.

5.4.1 Performance of textual LaTeX edit recommendation. Table 3 presents the two metrics score of different methods for modifying post sentences. Our MathLatexEdit achieves the best overall

Table 3. The Performance of different methods for textual LaTeX edit models

Models	BLEU score	GLEU score
SMT	59.22	52.64
Seq2seq	78.60	72.91
MathLatexEdit without sentence normalization	80.41	74.69
MathLatexEdit	82.30	76.57

Table 4. Examples of textual LaTeX edit with different methods

Original sentence	Our model	Seq2seq	SMT
1 formula: $y + py = px - 2p$ for which value (s) of p	formula: $y + py = px - 2p$ for which value (s) of p	formula: $y + py = px - 2p$ for which value (s) of p	formula: $y + py = px - 2p$ for which value (s) of p
2 'i' is part of the ratio	i is part of the ratio	i is part of the ratio	'i' is part of the ratio
3 we have seen also that the primitives $f(x, y)$	we have seen also that the primitives $f(x, y)$	we have seen also that the starts $f(x, y)$	we have seen also that the primitives $f(x, y)$
4 can some one explain $f(n) = 10 \cdot \log(n)$	can some one explain $f(n) = 10 \cdot \log(n)$	can some one explain $f(n) = 10 \cdot \log(n)$	can some one explain $f(n) = 10 \cdot \log(n)$

Table 5. The Performance of different methods with $test_{synthesize}$ and $test_{real}$ for visual LaTeX edit

Models	BLEU score	Image Similarity
INFTY($test_{synthesize}$)	67.72	50.21
INFTY($test_{real}$)	47.24	43.12
WYGIWYS($test_{synthesize}$)	89.92	90.22
WYGIWYS($test_{real}$)	70.21	77.21
MathLatexEdit($test_{synthesize}$) without image similarity	90.32	91.45
MathLatexEdit($test_{real}$) without image similarity	73.21	79.36
MathLatexEdit($test_{synthesize}$)	91.78	92.23
MathLatexEdit($test_{real}$)	74.71	81.61

result with the average BLEU score as 82.30, GLEU score as 76.57 which is 4.7%, 5.0% higher than Seq2seq model and 38.97%, 45.46% higher than SMT. The improvement in the two scores by our model represents a significant improvement over the two baseline methods. Note that in more detail, our model achieve 83.74 BLEU score, 77.41 GLEU score in formula latexification and 81.63 BLEU score, 76.03 GLEU score in LaTeX revision.

To qualitatively understand the strengths and weaknesses of different methods, we analyzed and compared the test results from three methods. Table 4 lists some representative examples in which MathLatexEdit outperforms the two baseline methods. Each row contains an original sentence and three edited sentences which are modified by different methods. SMT can edit some domain-specific words (e.g., $f(x,y)$ to $f(x,y)$ in 3rd example). But it often preserves the original sentences that should be edited. For instance, SMT fails to convert "p" into " p " in 1st example, fail to convert "i" into " i " in 2nd example and fail to convert "*" and "log" into " \cdot " and " \log " in 4th example. Therefore, SMT does not work well for minor math-related changes in post edits because it cannot fully utilize the context information.

Seq2seq works better than SMT, but still fails to edit some tokens. For example, "i" is incorrectly converted to " i " with an additional "i" in 2nd example and "primitives" is incorrectly converted into "starts" in 3rd example, "primitives" is incorrectly converted into "starts" in 3rd example, and it fails to convert "*" into " \cdot " in 4th example. The reason may be that Seq2seq incorrectly learns some biased knowledge from the training dataset and cannot handle the rare patterns. However, our MathLatexEdit can avoid the incorrect modification in sentence tokens with sentence normalization and provide better results in all the examples.

The ablation study of our model without sentence normalization shows that the domain-specific normalization contributes 2.3% and 2.5% improvement than the vanilla model in BLEU and GLEU

score. By analyzing low-quality recommendations by our model, we find four main reasons why our recommendation does not match the ground truth. First, some sentences are edited to add more information which is beyond the context of a sentence, such as editing "i need to find weight for x and y" to "I need to find weight w_1 and w_2 ". Our current model considers only the local context of a sentence. To support such complicated edits, the broader context of the sentence (i.e., previous and subsequent sentences) need to be considered in the future.

Second, our model may provide better results compared with the ground truth. For example, our model edit the sentence "if $f \circ f$ is differentiable, then $f \circ f \circ f$ is differentiable?" to "if $f \circ f$ is differentiable, then $f \circ f \circ f$ is differentiable?". While the ground truth is "if $f \circ f$ is differentiable, then $f \circ f \circ f$ is differentiable?". Compared with the ground truth, our model not only convert " \circ " to " \circ ", but also correctly add "\$" to " $f \circ f$ ".

Third, different users may have different opinions regarding what should or should not be edited. For example, some users will edit the sentence "what i know so far is $s = (y_2 - y_1) / (x_2 - x_1)$ " to "what I know so far is $s = \frac{y_2 - y_1}{x_2 - x_1}$ ". However, many other users will not do that. Many revert-back edits we see when collecting original-edited sentences are the evidence of such different opinions. Different editing opinions often result in non-obvious editing patterns, which machine learning techniques cannot effectively encode.

Fourth, the sentence length is a crucial factor influencing the performance of the model. Our model sometimes cannot fully correct all the errors in long LaTeX sequence. For example, so $k = 0.5 \sqrt{-16} = 2i$ is the solution is converted into so $k = 0.5 \sqrt{-16} = 2i$ is the solution. However, our model fails to convert $0.5 \sqrt{-16} = 2i$ to $0.5 \sqrt{-16} = 2i$ and e^{2ix} to e^{2ix} . To support such long sequence edits, splitting the long-length post sentence need to be considered in the future.

5.4.2 Performance of visual LaTeX edit recommendation. Table 5 presents the two metrics score of different methods for converting formula images to LaTeX formula in the $test_{synthesize}$ dataset of 50,000 image-formula pairs and $test_{real}$ of 100 image-formula pairs.

In $test_{synthesize}$, our MathLatexEdit achieves the best overall result with the average BLEU score as 91.78 and image similarity as 92.23. Due to the limitation of conventional image processing, INFTY gets the worst performance among all three models. The Deep learning based model, WYGIWYS has much better performance than INFTY. But MathLatexEdit can still achieve 2.0% and 2.2% boost in BLEU score and image similarity. In $test_{real}$, all three models show slightly worse performance compared with $test_{synthesize}$, as it is a more challenging task. MathLatexEdit can still outperform the other two baselines with reasonably good performance in all metrics. Adding re-ranking based on the image similarity during inferring helps our model improve 1.6% BLEU score in synthesized data and 2.8% image similarity score in real data.

To qualitatively understand the strengths and weaknesses of different methods, we randomly selected 600 results from three methods for a detailed comparison. Fig 6 lists some representative examples in which MathLatexEdit outperforms the two baseline methods. Each row contains an original formula image and three formula images rendered by the predicted LaTeX sequences by three models. INFTY does not work well for most cases as its rule-based methods do not scale well. It first segments the characters in the image and then recognizes each character by comparing it with candidates in their database. But there might be exceptions in each step and the errors in one step will be further amplified in the consecutive steps. For example, it cannot segment the f_1 and $'$ from f_1' in Fig 6 (a). The deep learning model, WYGIWYS behaves well in most cases, but it still makes mistakes especially for formulas with very fine-grained information. For example, it

Original images	INFTY	WYGIWYS	Our Model
(a) $f'_1, f'_2 : Y \rightarrow Z^X$	$f_1 f_2 Y \rightarrow Z^\lambda$	$f'_1, f'_2 : Y \rightarrow Z^X$	$f'_1, f'_2 : Y \rightarrow Z^X$
(b) $f_2(z) = \frac{-i((-iz)+1)}{(-iz)-1} = \frac{-z-i}{-iz-1}$	$f_1 f_2 Y \rightarrow Z^{\forall p(z)} = \frac{-i((-iz)+1)}{(-iz)-1} = \frac{-z-i}{-iz-1}$	$f_2(z) = \frac{-i((-i_z)+1)}{(-i_z)-1} = \frac{-z-i}{-i_z-1}$	$f_2(z) = \frac{-i((-iz)+1)}{(-iz)-1} = \frac{-z-i}{-iz-1}$
(c) $a-1+(a-1)e^{i\theta} = (a-i)(1+e^{i\theta})$	empty	$a-1+(a-1)e^\theta = (a-i)(1+e^\theta)$	$a-1+(a-1)e^{i\theta} = (a-i)(1+e^{i\theta})$
(d) $\sqrt{(x_1-a)^2+(x_2-a)^2} = b$	$(\text{REJECT})\neg(\text{REJECT}_3\text{REJECT})^\supset$	$\sqrt{(x_1-\alpha)^2+(x_2-\alpha)^2} = b$	$\sqrt{(x_1-\alpha)^2+(x_2-\alpha)^2} = b$

Fig. 6. Examples of converted LaTeX formulas by different models

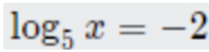
(a)	$\int_0^{2\pi} \exp[(r \cos(x) + s \sin(x))] \cdot \cos[(a \cos(x) + b \sin(x))] dx$	➡	$\int_0^{2\pi} \exp[(r \cos(x) + s \sin(x))] - \cos(a \cos(x) + b \sin(x)) dx$
(b)	$\widehat{a} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \vec{y}$	➡	$\widehat{a} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \vec{y}$
(c)		➡	$\log 5x = -2$

Fig. 7. Examples of wrongly predicted LaTeX formulas by MathLatexEdit

incorrectly predicts “iz” as “i_z” in Fig 6(b) and “e^{iθ}” as “e^θ” in Fig 6(c), as these characters are too close to each other and relatively small compared with the whole screenshot.

To analyze the low-quality recommendation by our model, we randomly selected some predictions which do not match the ground truth in both *test_{synthesize}* and *test_{real}*. According to our observation, we summarize three reasons for those erroneous predictions as Fig 7.

First, some input images have long or complex formulas, which makes it difficult to precisely predict the whole formula. For example, Fig 7.(a) is one formula image whose formula has a long length and complex structure. Although our predictions are right for most tokens, we still misclassify “.” as “-”, “(” and “)” as “[” and “]”. The longer the formula is, the more difficult we predict an exact match.

Second, some tokens in input formula images have some rare characters. For example, the special combination of →, ^ and a as \widehat{a} in Fig 7.(b) is rarely used. In the generated formula image, all other tokens except \widehat{a} are correctly predicted, but \widehat{a} is classified as \widehat{a} . In our collected 2,068,744 math LaTeX sequences from Mathematics Stack Exchange, $\widehat{}$ only appears once, resulting in the miss in MathLatexEdit.

Third, some input formula screenshots are too blurred or with some noisy background. For example, Fig 7.(c) is a screenshot taken from a book with low resolution and gray background. Most part of the generated LaTeX formula matches the ground truth, but there are still some minor errors like misclassifying “log₅” as “log 5”.

6 REAL-WORLD EVALUATION OF EDITS

Having established confidence in the quality of modified posts by MathLatexEdit, we further investigated whether the recommended edits by MathLatexEdit can actually assist post editors especially those with little editing experience or expertise to successfully improve the quality of

Table 6. The comparison of the experiment and control group. * denotes $p < 0.01$

Post ID	Time cost (s)		Precision		Recall		Satisfactoriness	
	CG	EG	CG	EG	CG	EG	CG	EG
1	345.6	56.4	62.41	90.24	74.21	91.62	3.33	4.50
2	373.6	60.2	70.21	92.92	70.53	89.48	3.50	5.00
3	424.8	70.2	61.42	94.75	75.47	93.70	3.83	4.83
4	483.4	88.6	71.23	93.27	78.64	94.57	3.50	4.50
5	262.2	60.6	63.42	92.88	69.83	90.29	3.67	4.83
6	382.4	47.8	60.74	89.22	71.42	90.58	3.67	5.00
Average	378.6	64.0*	68.70	92.21*	73.35	91.70*	3.58	4.78*

mathematics posts in practice. We conducted a user study and a field study in this section to further evaluate the usefulness of our model for assisting domain-specific edits.

6.1 Usefulness Evaluation

6.1.1 Procedures for User Study. We randomly selected 50 posts to manually check its needed edits. We asked an experienced research staff (not involved in the study) to edit these posts and collected corresponding edits as the groundtruth. Note that these posts are selected outside our training corpus (i.e., date after May 1, 2020) to avoid potential bias. Among 50 posts, 14 of them require math-related edits and we further selected 6 needing different kinds of edits for this user study.

We recruited ten PhD students and research assistants from our school. According to pre-study background survey, all participants knew Mathematics Stack Exchange and were familiar with using LaTeX. The study involves the two groups of five participants: the experimental group P_1, P_2, P_3, P_4, P_5 who do the post editing with our tool, and the control group $P_6, P_7, P_8, P_9, P_{10}$ who start from scratch. Each pair of participants $\langle P_x, P_{x+5} \rangle$ have comparable experience in using Mathematics Stack Exchange and Latex so that the experimental group has similar expertise to the control group in total. Note that we do not ask participants to edit half of the posts with our tool while the other half without assisting tools to avoid potential tool bias.

We gave participants a detailed explanation of our tool for helping them understand the results. Participants were required to edit posts with/without our tool and have up to 10 minutes for each post. We recorded the time used to edit each post for every participant. After each post editing, participants were asked to rate how satisfied they are with their edits in five-point Likert scale [38] (1: not satisfied at all and 5: highly satisfied). Compared with the groundtruth, we calculated the precision and recall for these post edits.

6.1.2 Results of User Study. Table 6 shows that participants in the experimental group spent less time (on average 64s versus 378.6s) in editing posts than the control group. It indicates that our tool helps editors save 82.85% editing time. In fact, the average time of the control group is underestimated, as 2 participants failed to complete at least one post within 10 minutes, which means that they may need more time in the real editing. In contrast, all participants in the experimental group finished all post editing within 3 minutes. With the help of our MathLatexEdit, the experimental group also obtained higher (34.22% and 25.02%) precision and recall score in terms of the edits. The experimental group rated 80% of their post edits as highly satisfactory (5 point), as opposed to 10% highly satisfactory by the control group. On average, the satisfactoriness scores for the experiment and control group are 4.78 versus 3.58.

We carried out the Mann-Whitney U test [25] (specifically designed for small samples) to understand the significance of the differences between the two groups. It suggests that our tool can

May 17	suggested	approved edit on How to cluster words using Jensen-Shannon Divergence?
May 17	suggested	approved edit on Elementary schools summation in denominator?
May 17	suggested	approved edit on Proof of Poisson Summation Formula
May 17	suggested	approved edit on Make r the Subject of the formula

Fig. 8. Example of MathLatexEdit being used to assist post editors

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad \longrightarrow \quad x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Fig. 9. Example of rejected MathLatexEdit generated revision

significantly help the experimental group edit posts faster ($p - value < 0.01$), with higher precision, recall and satisfactoriness score ($p - value < 0.01$).

We believe that the better performance of the experimental group is due to the assistance of our MathLatexEdit which gives participants a reliable starting point for editing. Guided by the recommended post edits, participants can easily identify the issues and finish the edit. 83.3% recommendations provided by our tool are directly accepted by the editors. Although there may be some mistakes in our tool’s recommendation, participants can easily revise them. Without the help of edit recommendation, the control group have to read the post thoroughly and determine where issues are and solve these issues from scratch, which results in the longer edit time and less satisfactory edit results. For example, some latexification edits within text like replacing $x-y$ to $x - y$ are frequently missed in the control group. The revision of complex latex formula such as $\langle x, x \rangle \cdot \langle y, y \rangle = |\langle x, y \rangle|^2 + |x \times y|^2$ is more error-prone in control group.

6.2 Real-world Assistance for Post Editors

We also conducted a small-scale field study, in which the first author who has no experience in Mathematics Stack Exchange acted as a novice post editor. We randomly selected 600 posts (106 are with images) after May 1, 2020 which are never used in our training dataset to avoid potential bias. Our model found that 112 posts need at least one LaTeX edit (including 95 posts with textual LaTeX edit, 22 posts with visual LaTeX edits). Among those 112 posts, we selected 80 posts (60 with only textual Latex edits, 20 with only visual Latex edits) which is of reasonable size, and manageable with human effort to manually submit the post edits. In Mathematics Stack Exchange, each question can have up to 5 tags to describe its topic. The 80 selected posts contain 203 tags in total (if the post is the answer, we took tags from its corresponding question) and 127 of these tags are unique. This indicates that the 80 selected posts cover a diverse set of mathematical topics. In fact, these posts contain many mathematical terms that are beyond the expertise of the first author. Within 80 selected posts, 69 (86.25%) of them just involve one edit while 11 (13.75%) of them involve multiple edits.

After proofreading the 80 posts edited by our MathLatexEdit, he submitted the post edits to the community for approval. For the 80 submitted post edits, 78 (97.5%) were accepted and 2 (2.5%) were

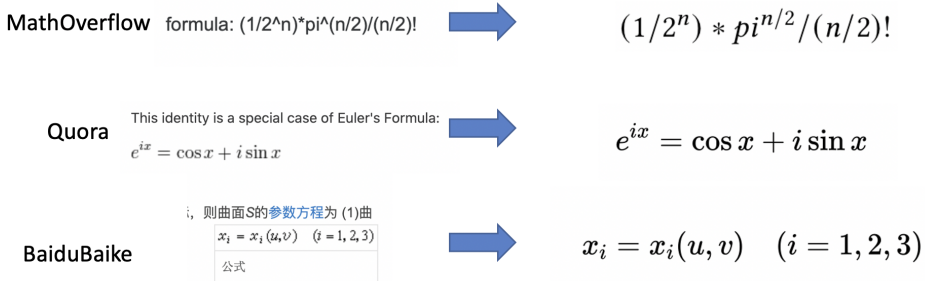


Fig. 10. Example of MathLatexEdit generated revision from other sites

rejected. And the two rejected ones are both single edits. The experiment results show that no matter there are single edits or multiple edits in one post, most of them are accepted by the community. Records of some edits are shown in Fig 8. For the two rejected edits, which are visual LaTeX edits, the original images have special notations in the formula. Although MathLatexEdit recommended the correct LaTeX sequences, the notations in the formulas images were missed. Figure 9 shows one example with yellow notation of some parts of the formula for illustrating the rejection. Since our revised LaTeX sequence misses the notation which is important for this post, the revision was not accepted. But for the 78 accepted post edits, the trusted contributors believed that they contained sufficient correct edits that significantly improved the post quality, and thus approved them. This real-world acceptance of our MathLatexEdit's proposed edits demonstrates the usefulness of MathLatexEdit in formula latexification, LaTeX revision and screenshot transcription for the Mathematics Stack Exchange Q&A collaborative site.

7 DISCUSSION AND FUTURE WORKS

In this section, we discuss the possible generalization of our MathLatexEdit approach for use to support collaboration on other Q&A sites, and also its potential impact to improve accessibility of online mathematics-related information.

7.1 Generalization of MathLatexEdit

This work examined collaborative editing patterns in Mathematics Stack Exchange, and we developed MathLatexEdit, a deep learning-based approach for latexifying formula, revising latex and transcribing screenshots to assist post owners and editors. Note that our data analysis method and deep learning approach are totally data driven, and not tied to any specific collaborative editing or quality control process used by particular community Q&A sites. In this work, we study only Mathematics Stack Exchange post edits. However, the input to our approach is essentially just a parallel corpus of original and edited text (see Figure 1). Therefore, we would expect that our data analysis method and deep learning approach could be applied to other mathematics-related Q&A sites.

There are also some other sites which contain mathematics formulas and also support collaborative editing, such as MathOverflow¹³ (math Q&A site for professional mathematicians in Stack Exchange network), Quora¹⁴ (largest general Q&A site), BaiduBaike¹⁵ (largest online encyclopedia created and edited by volunteers in China). We randomly selected 15 posts that require formula

¹³<https://mathoverflow.net/>

¹⁴<https://www.quora.com/>

¹⁵<https://baike.baidu.com/>

latexification, latex revision or screenshot transcription from each site and edited them with our tool, as shown in Fig 10. Since BaiduBaiKe requires certain reputation score for submission while we failed to reach that score, we only submitted our revisions in MathOverflow and Quora. We submitted our revisions to corresponding sites at 5 Oct, 2020. MathOverflow accepted 15 edits, and Quora accepted 10 edits with the other 5 pending. These results demonstrate the generalization of our MathLatexEdit approach, and also the possibility of MathLatexEdit to be adapted to use for these sites.

However, it is still an open question whether our approach can perform well in a large-scale, live deployment on mathematics Q&A sites. Several issues require further investigation, such as: how to integrate our recommendation in the Q&A and collaborative editing processes; can post owners and editors easily solve errors made in our generated LaTeX; how does MathLatexEdit usage impact the post owners and post editors' behavior; and how does the suggested edits of MathLatexEdit lead to changes in people's perceptions of the math formula.

7.2 Implications of MathLatexEdit to Accessible Math

Converting a posted mathematics formula screenshot to Latex brings many benefits to the site. First, as shown in Fig 6, transcribing the mathematics formula screenshot makes the site easier to read for everyone, as users can zoom in or out for a clearer view without blurring the formula. Such readability can help other users better understand the question, resulting in higher possibility of responses.

Second, this machine-readability also makes the text easier to register and index by search-engine crawlers. This in turn makes the post easier to search for and easier to be found by other users, and this, therefore, makes the post more useful for a broader cross-section of internet users.

The screenshot transcribing feature would also make the questions more accessible to users with vision impairment, where the formula can be read by screen reader. The users with visual impairments, ranging from difficulty reading all the way to complete blindness, need to be able to use the internet and study maths [9]. When visiting the website, they mainly rely on assistive technology such as screen readers which can only read the text, rather than the image. Using proper typesetting (including good use of Markdown and Mathjax) provides additional HTML syntax that those assistive technologies can use to give a more meaningful account of the content on the page.

Although MathLatexEdit makes it easier for users to convert an image to LaTeX sequence, not all users will use it or follow the guidelines of Mathematics Stack Exchange. Indeed, outside Mathematics Stack Exchange there are still many screenshots of mathematics formulas on different sites. These inaccessible screenshots put a major barrier for users with vision impairment in learning and using mathematics online, resulting in degradation of their education equality. MathLatexEdit can help with converting a digital image into a LaTeX textual mathematics formula. Physical books are another kind of important learning resources for those special students, our model has important potential to be adapted for mathematics formula recognition from physical or image-based electronic book pictures.

However, note that what we generate is only the LaTeX sequences, which are not very user friendly for most end users. A simple division is written as “ $\frac{\text{numerator}}{\text{denominator}}$ ” as an example in LaTeX, and this is much more complicated than how we speak about the mathematics orally. Therefore, in future work we want to develop a set of rules to further convert the generated LaTeX from MathLatexEdit into the natural language of mathematics formula to make it more accessible for novice users.

8 CONCLUSION

In this paper, we carried out an empirical study of historical collaborative editing data on the Mathematics Stack Exchange. Our results showed that collaborative editing is widely used in Mathematics Stack Exchange, which includes three domain-specific editing use cases as formula latexification, latex revision and screenshot transcription. Due to the difficulty of the conversion, we designed `MathLatexEdit`, a deep learning-based approach that automatically revises the math-related content in post. `MathLatexEdit`'s recommendations can assist post owners and editors in improving dissemination of mathematics knowledge via the Q&A site. Our evaluation through use of large-scale datasets demonstrates the quality of the edited LaTeX sequence by `MathLatexEdit`. The edit recommendations from `MathLatexEdit` for a selection of real-world posts were accepted by experienced users of Mathematics Stack Exchange, further showing the usefulness of our tool. We discussed the potential benefits of our `MathLatexEdit` post-edit recommendation approach for post owners/editors as well as different platforms. However, deploying our approach on these sites may have complicated impacts on social process and collaborative editing, which need further study in the future.

ACKNOWLEDGMENTS

Ma is supported by a Faculty of IT PhD scholarship. Grundy and Khalajzadeh are supported by ARC Laureate Fellowship FL190100035.

REFERENCES

- [1] 2020. About wikiHow. [https://math.stackexchange.com/help/how-to-answer](https://www.wikihow.com/wikiHow>About-wikiHow. Accessed May 26, 2020.
[2] 2020. How do I write a good answer? <a href=). Accessed May 26, 2020.
- [3] 2020. How to Ask. <https://math.stackexchange.com/questions/ask/advice>. Accessed May 26, 2020.
- [4] 2020. MathJax basic tutorial and quick reference. <https://math.meta.stackexchange.com/questions/5020/mathjax-basic-tutorial-and-quick-reference>. Accessed Oct 16, 2020.
- [5] 2020. The Objective Revision Evaluation Service. <https://ores.wikimedia.org/>. Accessed May 25, 2020.
- [6] 2020. Stack Exchange. <https://stackexchange.com/sites#>. Accessed May 28, 2020.
- [7] 2020. Welcome to Mathematics Stack Exchange. <https://math.stackexchange.com/tour>. Accessed May 26, 2020.
- [8] 2020. Why can people edit my posts? How does editing work? <https://math.stackexchange.com/help/editing>. Accessed May 26, 2020.
- [9] Nancy Alajarmeh. 2012. Doing math: mathematics accessibility issues. In *Proceedings of the International Cross-Disciplinary Conference on web accessibility (W4A '12)*. ACM, 1–2.
- [10] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.
- [11] Kaibo Cao, Chunyang Chen, Sebastian Baltes, Christoph Treude, and Xiang Chen. 2021. Automated Query Reformulation for Efficient Search based on Query Logs From Stack Overflow. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*. IEEE, 1273–1285.
- [12] Davide Cervone. 2012. MathJax: A Platform for Mathematics on the Web. *Notices of the American Mathematical Society* 59, 02 (2012), 1.
- [13] Chunyang Chen, Xi Chen, Jiamou Sun, Zhenchang Xing, and Guoqiang Li. 2018. Data-driven proactive policy assurance of post quality in community q&a sites. *Proceedings of the ACM on human-computer interaction* 2, CSCW (2018), 1–22.
- [14] Chunyang Chen, Ting Su, Guozhu Meng, Zhenchang Xing, and Yang Liu. 2018. From ui design image to gui skeleton: a neural machine translator to bootstrap mobile gui implementation. In *Proceedings of the 40th International Conference on Software Engineering*. 665–676.
- [15] Chunyang Chen and Zhenchang Xing. 2016. Mining technology landscape from stack overflow. In *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. 1–10.
- [16] Chunyang Chen, Zhenchang Xing, and Lei Han. 2016. Techland: Assisting technology landscape inquiries with insights from stack overflow. In *2016 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. IEEE, 356–366.
- [17] Chunyang Chen, Zhenchang Xing, and Yang Liu. 2017. By the community & for the community: a deep learning approach to assist collaborative editing in q&a sites. *Proceedings of the ACM on Human-Computer Interaction* 1, CSCW (2017), 1–21.

- [18] Chunyang Chen, Zhenchang Xing, and Ximing Wang. 2017. Unsupervised software-specific morphological forms inference from informal discussions. In *2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE)*. IEEE, 450–461.
- [19] Jieshan Chen, Chunyang Chen, Zhenchang Xing, Xiwei Xu, Liming Zhut, Guoqiang Li, and Jinshui Wang. 2020. Unblind your apps: Predicting natural-language labels for mobile gui components by deep learning. In *2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE)*. IEEE, 322–334.
- [20] Xiang Chen, Chunyang Chen, Dun Zhang, and Zhenchang Xing. 2019. Sthesaurus: Wordnet in software engineering. *IEEE Transactions on Software Engineering* (2019).
- [21] Joohee Choi and Yla Tausczik. 2018. Will Too Many Editors Spoil The Tag? Conflicts and Alignment in Q&A Categorization. *Proceedings of the ACM on Human-Computer Interaction* 2, CSCW (2018), 1–19.
- [22] Shamil Chollampatt and Hwee Tou Ng. 2018. Neural Quality Estimation of Grammatical Error Correction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, 2528–2539. <https://doi.org/10.18653/v1/D18-1274>
- [23] Marcel. author Danesi. 2016. *Learning and Teaching Mathematics in The Global Village Math Education in the Digital Age* (1st ed. 2016. ed.).
- [24] Yuntian Deng, Anssi Kanervisto, and Alexander M Rush. 2016. What you get is what you see: A visual markup decompiler. *arXiv preprint arXiv:1609.04938* 10 (2016), 32–37.
- [25] Michael P Fay and Michael A Proschan. 2010. Wilcoxon-Mann-Whitney or t-test? On assumptions for hypothesis tests and multiple interpretations of decision rules. *Statistics surveys* 4 (2010), 1–39. <https://doi.org/10.1214/09-ss051>
- [26] Denae Ford, Kristina Lustig, Jeremy Banks, and Chris Parnin. 2018. " We Don't Do That Here" How Collaborative Editing with Mentors Improves Engagement in Social Q&A Communities. In *Proceedings of the 2018 CHI conference on human factors in computing systems*. 1–12.
- [27] Sa Gao, Chunyang Chen, Zhenchang Xing, Yukun Ma, Wen Song, and Shang-Wei Lin. 2019. A neural model for method name generation from functional description. In *2019 IEEE 26th International Conference on Software Analysis, Evolution and Reengineering (SANER)*. IEEE, 414–421.
- [28] Utpal Garain, BB Chaudhuri, and Adrish Ray Chaudhuri. 2004. Identification of embedded mathematical expressions in scanned documents. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004., Vol. 1*. IEEE, 384–387.
- [29] Roman Grundkiewicz and Marcin Junczys-Dowmunt. 2018. Near Human-Level Performance in Grammatical Error Correction with Hybrid Machine Translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 284–290. <https://doi.org/10.18653/v1/N18-2046>
- [30] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.
- [31] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4700–4708.
- [32] Karen Embry Jenlink. 2006. Math Education. , 647–651 pages.
- [33] Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2016. Phrase-based machine translation is state-of-the-art for automatic grammatical error correction. *arXiv preprint arXiv:1605.06353* (2016).
- [34] Jean Lave and Etienne Wenger. 1999. Legitimate peripheral participation. *Learners, learning and assessment, London: The Open University* (1999), 83–89.
- [35] Vladimir I Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, Vol. 10. 707–710.
- [36] Guo Li, Tun Lu, Xianghua Ding, and Ning Gu. 2016. Predicting Collaborative Edits of Questions and Answers in Online Q&A Sites. *Journal of Internet Technology* 17 (2016), 1187–1194.
- [37] Guo Li, Haiyi Zhu, Tun Lu, Xianghua Ding, and Ning Gu. 2015. Is it good to be like Wikipedia? Exploring the trade-offs of introducing collaborative editing model to Q&A sites. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*. 1080–1091.
- [38] Rensis Likert. 1932. A technique for the measurement of attitudes. *Archives of psychology* (1932).
- [39] Suyu Ma, Zhenchang Xing, Chunyang Chen, Cheng Chen, Lizhen Qu, and Guoqiang Li. 2019. Easy-to-deploy API extraction by multi-level feature embedding and transfer learning. *IEEE Transactions on Software Engineering* (2019).
- [40] Lena Mamykina, Bella Manoim, Manas Mittal, George Hripsak, and Björn Hartmann. 2011. Design lessons from the fastest q&a site in the west. In *Proceedings of the SIGCHI Conference on human factors in computing systems (CHI '11)*. ACM, 2857–2866.
- [41] Jessica Middendorf. 2018. Increasing Retention through Math Study Skills. <http://search.proquest.com/docview/2061549897/>
- [42] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

- [43] Tomoya Mizumoto and Yuji Matsumoto. 2016. Discriminative reranking for grammatical error correction with statistical machine translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 1133–1138.
- [44] Leydi Viviana Montoya, Athen Ma, and Raúl J Mondragón. 2013. Social achievement and centrality in MathOverflow. In *Complex Networks IV*. Springer, 27–38.
- [45] Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. 2016. GLEU without tuning. *arXiv preprint arXiv:1605.02592* (2016).
- [46] Daniel Ortiz-Martinez, Ismael Garcia-Varea, and Francisco Casacuberta. 2005. Thot: a toolkit to train phrase-based statistical translation models. *Tenth Machine Translation Summit. AAMT, Phuket, Thailand, September* (2005).
- [47] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, 311–318. <https://www.aclweb.org/anthology/P02-1040>
- [48] Masakazu Suzuki, Fumikazu Tamari, Ryoji Fukuda, Seiichi Uchida, and Toshihiro Kanahori. 2003. INFTY: an integrated OCR system for mathematical documents. In *Proceedings of the 2003 ACM symposium on document engineering (DocEng '03)*. ACM, 95–104.
- [49] Yla R Tausczik, Aniket Kittur, and Robert E Kraut. 2014. Collaborative problem solving: A study of mathoverflow. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*. 355–367.
- [50] Hashim M Twaakyondo and Masayuki Okamoto. 1995. Structure analysis and recognition of mathematical expressions. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, Vol. 1. IEEE, 430–437.
- [51] Andrew W Vargo and Shigeo Matsubara. 2016. Editing Unfit Questions in Q&A. In *2016 5th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI)*. IEEE, 107–112.
- [52] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.
- [53] Jian Wang, Yunchuan Sun, and Shenling Wang. 2019. Image To Latex with DenseNet Encoder and Joint Attention. *Procedia computer science* 147 (2019), 374–380.
- [54] Xu Wang, Chunyang Chen, and Zhenchang Xing. 2019. Domain-specific machine translation with recurrent neural network for software localization. *Empirical Software Engineering* 24, 6 (2019), 3514–3545.
- [55] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*. 2048–2057.
- [56] Zheng Yuan and Ted Briscoe. 2016. Grammatical error correction using neural machine translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 380–386.
- [57] Zheng Yuan, Ted Briscoe, and Mariano Felice. 2016. Candidate re-ranking for SMT-based grammatical error correction. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*. 256–266.

Received October 2020 ; revised April 2021 ; accepted July 2021