Capstone Proposal

Math Formula Images To LaTeX

1. Domain Background

Mathematics is a foundational subject that permeates numerous fields, including science, technology, education, and engineering. Mathematical formulas are crucial for describing algorithms, illustrating ideas, clarifying complex concepts, and ensuring consistency in communication. Representing these formulas effectively and accurately is essential for research, teaching, and documentation.

LaTeX, a widely used markup language, is a standard tool for typesetting mathematical content professionally. From research papers to educational materials, LaTeX provides precision and clarity, making it indispensable in academia and beyond. However, despite its advantages, creating LaTeX representations of mathematical formulas is a daunting task, especially for non-experts. Even for professionals, typing lengthy and intricate formulas can lead to errors, frustration, and inefficiencies in editing and debugging.

Objectives: Build a robust model that recognizes mathematical formulas from images and translates them to a string of LaTeX markup code.

Problem in the Domain

As LaTeX usage expands beyond academia to include websites, blogs, and educational platforms, the demand for tools that simplify its use has grown. One significant challenge is converting mathematical formulas from their visual representation (e.g., handwritten notes or printed images) into LaTeX code. Manual conversion is time-consuming and error-prone, particularly for complex or lengthy formulas. This issue is exacerbated for beginners, who often struggle to learn LaTeX syntax and structure. Providing a solution that bridges the gap between visual formulas and LaTeX sequences would not only enhance productivity but also make LaTeX more accessible to a broader audience.

Relevance of Historical Context

The need for automated formula recognition has been recognized for decades. Early attempts focused on optical character recognition (OCR) for general text, but the structured and hierarchical nature of mathematical expressions posed unique challenges. Over time, specialized approaches have emerged, leveraging advancements in computer vision and natural language processing to address these challenges. Recently, encoder-decoder architectures have proven effective in handling structured data like mathematical expressions, enabling significant progress in this domain.

Academic Research

The advent of deep neural networks has replaced classical methods with encoder decoder architectures and brought about good results and real success in the field of Computer Vision & Natural Language Processing. Methods using encoder decoder architectures such as [1] propose a neural encoder-decoder with a coarse-to-fine attention mechanism. The authors use a Convolutional Neural Network (CNN) encoder to extract features from the image and an Recurrent Neural Network (RNN) decoder implements a conditional language model over the vocabulary. [2] introduces a neural transducer model with visual attention, which uses a CNN as an encoder and an RNN as a decoder, combined with beam search during inference. [3] proposes a method that includes a CNN combined with positional encoding used in the encoder to extract features. The features are augmented with 2D positional encoding before being unfolded into a vector and fed into Long Short Term Memory (LSTM) decoder to translate into a sequence of LaTeX tokens. [4] proposes a model that applies a Transformer-based encoder decoder architecture. The encoder uses a Vision Transformer (ViT) and takes inspiration from machine translation to apply to the img2latex task. Additionally, this method combines the use of a YOLO model [5] for the preprocessing step of separating single-line formulas from multi-line formulas to improve the model's accuracy. BTTR [6] and ABM [7] methods introduce a novel bidirectional training strategy with the aim of learning LaTeX sequences from left-to-right and right-to-left directions on the RNN decoder to solve the lack of coverage problem [8]. However, this leads to more parameters and longer training time. Inspired by the coverage mechanism in RNN, CoMER [9] proposes a model that improves the Transformer's shortcomings regarding the lack of coverage problem. It uses an Attention Refinement Module (ARM) to refine attention weights with past alignment information without hurting its parallelism and performs better than the vanilla transformer decoder and RNN decoder in the HMER task.

2. Problem Statement

The primary problem is the inefficiency and difficulty in converting mathematical formulas from visual formats (handwritten or printed images) into LaTeX code. This process is currently manual, time-intensive, and prone to errors, especially for complex or lengthy formulas. Beginners face significant barriers in learning and using LaTeX effectively, and even experienced users often encounter challenges in editing and debugging intricate LaTeX sequences.

This problem is well-defined and quantifiable. The task can be measured by evaluating the accuracy of automated systems in converting image-based mathematical formulas into LaTeX code. Relevant metrics include BLEU scores for syntactic accuracy, Character Error Rate (CER) for text-level accuracy, and overall formula correctness. A potential solution involves developing a machine learning model, such as an encoder-decoder architecture, that leverages a large-scale dataset of formula-image pairs to perform this conversion efficiently. This approach is replicable, as the datasets, models, and evaluation metrics can be standardized and reused for future research and development.

3. Solution Statement

The proposed solution involves developing a model based on the Transformer encoder-decoder architecture to address the challenge of converting images containing mathematical formulas into LaTeX code. This end-to-end approach uses Swin Transformer for the encoder and Transformer for the decoder, leveraging their ability to process complex input images and generate accurate output sequences. By training on a large-scale dataset of self-collected and processed image-text (handwritten and printed formulas).

Key features of the solution include:

- Data Normalization: Pre-process formulas into standardized LaTeX syntax using the KaTeX parser, reducing ambiguity in representation.
- Data Augmentation: Enhance the robustness of the model through various image transformations.

The performance of the solution can be quantified using metrics such as BLEU score, Edit Distance (ED), and Word Error Rate (WER), demonstrating the effectiveness of the solution. The proposed model is reproducible as it is built on publicly available datasets and open source frameworks, ensuring reproducibility for future research.

4. Datasets and Inputs

The project uses a large-scale dataset consisting of approximately 3.4 million image-text pairs, encompassing both handwritten and printed mathematical formulas. This dataset is the largest of its kind, ensuring robust model training and generalization across diverse formula types. The dataset is divided as follows:

Printed Mathematical Formulas:

- Im2latex-100k dataset [10]
- I2L-140K Normalized dataset and Im2latex-90k Normalized dataset [11]
- Im2latex-170k dataset [12]
- Im2latex-230k dataset [13]
- Other [14]

Handwritten Mathematical Formulas:

- CROHME dataset [15, 16, 17]
- Aida Calculus Math Handwriting Recognition Dataset [18]
- Handwritten Mathematical Expression Convert LaTeX dataset [14]

Preprocessing

To address issues of polymorphic ambiguity—where a single mathematical formula can have multiple LaTeX representations—the dataset is preprocessed using a KaTeX parser [19]. The raw LaTeX strings are

converted into abstract syntax trees and normalized to create a standardized format. This ensures consistency and reduces errors during training.

5. Benchmark Model

To provide a context for evaluating the proposed solution, a benchmark model is utilized. Specifically, the Im2latex-100k dataset is used for experiments on the Printed Mathematical Expression Recognition task. This dataset contains approximately 100,000 real-world mathematical expressions rendered from public papers on the arXiv.org server, making it a popular choice for PMER research. The test set consists of 10,285 samples, enabling comprehensive evaluation.

In addition, the CROHME dataset is used to demonstrate the effectiveness of the model on the Handwritten Mathematical Expression Recognition task. This dataset is a large open collection for handwritten mathematical expressions. The test set comprises three versions: CROHME 2014 (986 samples), CROHME 2016 (1,147 samples), and CROHME 2019 (1,199 samples).

These datasets ensure a robust evaluation across both handwritten and printed domains.

6. Evaluation Metrics

To quantify the performance of both the benchmark and the proposed solution models, the following metrics are employed:

- BLEU Score: Measures the similarity between model-generated LaTeX code and reference translations using n-gram overlap. A higher BLEU score indicates greater syntactic and semantic alignment.
- Edit Distance (ED): Also known as Levenshtein distance, calculates the minimum number of edit operations (insertion, deletion, substitution) needed to transform the predicted LaTeX code into the reference. Accuracy is calculated as , where lower edit distance indicates better performance.
- Exact Match (EM): Evaluates the percentage of predictions that exactly match the reference LaTeX code. This metric highlights how often the model outputs are 100% accurate.
- Word Error Rate (WER): Based on edit distance, calculates the error rate as a fraction of substitutions, insertions, and deletions divided by the total number of words in the reference. Lower WER indicates higher accuracy.
- Expression Recognition Rate (ExpRate): Specifically designed for mathematical expressions, evaluates the percentage of lines with an edit distance of zero. This metric emphasizes exact recognition of entire formulas

7. Project Design

Workflow

- Data Collection and Preprocessing:
    - Aggregate printed and handwritten mathematical expression datasets (e.g., Im2latex-100k, CROHME datasets).
    - Normalize LaTeX expressions using a KaTeX parser to handle polymorphic ambiguity.
    - Perform data augmentation on handwritten samples to improve generalization.

- Training Model:
    - Implement a Transformer-based encoder-decoder architecture with Swin Transformer as the encoder and a Transformer as the decoder.
    - Train the model on preprocessed datasets, applying techniques like learning rate scheduling, dropout, and gradient clipping.

- Evaluation Model:
    - Test the trained model using standard datasets such as Im2latex-100k and CROHME.
    - Evaluate performance using BLEU scores, Edit Distance (ED), Exact Match (EM), and Word Error Rate (WER).

- Deployment: Deploy the trained model to a Streamlit-based web interface, enabling users to upload images and receive LaTeX code as output.

References

[1] Yuntian Deng et al. “Image-to-Markup Generation with Coarse-to-Fine Attention”. In: Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, 2017, pp. 980–989.

[2] Sumeet S. Singh. “Teaching Machines to Code: Neural Markup Generation with Visual Attention”. In: CoRR abs/1802.05415 (2018). arXiv: 1802.05415. url: http://arxiv.org/abs/1802.05415.

[3] Zelun Wang and Jyh-Charn Liu. “Translating math formula images to LaTeX sequences using deep neural networks with sequence-level training”. In: Int. J. Document Anal. Recognit. 24.1 (2021), pp. 63–75. doi: 10.1007/ S10032-020-00360-2. url: https://doi.org/10.1007/s10032-020-00360-2.

[4] Mingle Zhou et al. “An End-to-End Formula Recognition Method Integrated Attention Mechanism”. In: Mathematics 11.1 (2023). issn: 2227-7390. doi: 10.3390/math11010177. url: https://www.mdpi.com/2227-7390/11/1/177

[5] Joseph Redmon et al. “You Only Look Once: Unified, Real-Time Object Detection”. In: CoRR abs/1506.02640 (2015). arXiv: 1506.02640. url: http://arxiv.org/abs/1506.02640.

[6] Wenqi Zhao et al. “Handwritten Mathematical Expression Recognition with Bidirectionally Trained Transformer”. In: CoRR abs/2105.02412 (2021). arXiv: 2105.02412. url: https://arxiv.org/abs/2105.02412.

[7] Xiaohang Bian et al. “Handwritten Mathematical Expression Recognition via Attention Aggregation based Bi-directional Mutual Learning”. In: CoRR abs/2112.03603 (2021). arXiv: 2112.03603. url: https://arxiv.org/abs/2112.03603.

[8] Jianshu Zhang et al. “Watch, attend and parse: An end-to-end neural net work based approach to handwritten mathematical expression recognition”. In: Pattern Recognit. 71 (2017), pp. 196–206. doi: 10.1016/J.PATCOG. 2017.06.017. url: https://doi.org/10.1016/j.patcog.2017.06.017.

[9] WenqiZhaoandLiangcai Gao. “CoMER:ModelingCoverage for Transformer Based Handwritten Mathematical Expression Recognition”. In: Computer Vision– ECCV 2022. Ed. by Shai Avidan et al. Cham: Springer Nature Switzerland, 2022, pp. 392–408. isbn: 978-3-031-19815-1.

[10] Yuntian Deng, Anssi Kanervisto, and Alexander M. Rush. “What You Get Is What You See: A Visual Markup Decompiler”. In: CoRR abs/1609.04938 (2016). arXiv: 1609.04938. url: http://arxiv.org/abs/1609.04938.

[11] Sumeet S. Singh. “Teaching Machines to Code: Neural Markup Generation with Visual Attention”. In: CoRR abs/1802.05415 (2018). arXiv: 1802.05415. url: http://arxiv.org/abs/1802.05415.

[12] https://www.kaggle.com/datasets/rvente/im2latex170k.

[13] https://www.kaggle.com/datasets/gregoryeritsyan/im2latex-230k.

[14] https://huggingface.co

[15] Mahshad Mahdavi et al. “ICDAR 2019 CROHME + TFD: Competition on Recognition of Handwritten Mathematical Expressions and Typeset Formula Detection”. In: 2019 International Conference on Document Anal ysis and Recognition, ICDAR 2019, Sydney, Australia, September 20-25, 2019. IEEE, 2019, pp. 1533–1538. doi: 10.1109/ICDAR.2019.00247. url: https://doi.org/10.1109/ICDAR.2019.00247.

[16] Harold Mouchère et al. “ICFHR 2014 Competition on Recognition of On Line Handwritten Mathematical Expressions (CROHME 2014)”. In: 14th International Conference on Frontiers in Handwriting Recognition, ICFHR 2014, Crete, Greece, September 1-4, 2014. IEEE Computer Society, 2014, pp. 791–796. doi: 10.1109/ICFHR.2014.138.

[17] Harold Mouchère et al. “ICFHR2016 CROHME: Competition on Recogni tion of Online Handwritten Mathematical Expressions”. In: 15th Interna tional Conference on Frontiers in Handwriting Recognition, ICFHR 2016, Shenzhen, China, October 23-26, 2016. IEEE Computer Society, 2016, pp. 607–612. doi: 10.1109/ICFHR.2016.0116.

[18] https://www.v7labs.com/darwin

[19] https://katex.org/