Hunter TEXNet | CSCI 350 Project Proposal

Alex Taradachuck Ralph "Blake" Venté

March 2020

Abstract

The OpenAI request for research spawned research in using Visual Neural Models for translating mathematical expressions into markup code. Since the debut, progress has been largely incremental. Singh 2018, Wang 2019, and Bender 2019 all attribute properties of their respect models to the biases in the corpus and its pre-processing. We present a new dataset and a pipeline for harvesting examples from source files hoping to minimize these biases.

1 Previous Work

The prospect of accurately transcribing mathematical expression into a markup representation is enticing because it opens the doors for bringing new life to old mathematical texts or those for which the source code is unavailable.

A Harvard project called *What you get is what you see* (WYGIWYS) by Deng, Kanervisto, and Rush 2016 documents strategies for machine translation of mathematical notation using an attention-based encoder-decoder neural model. Notably, the researchers were interested in the influence of markup alone on the efficacy of a model — without providing explicit information about underlying grammars (Deng, Kanervisto, and Rush 2016, p. 1).

Introduced in 2016 by Deng, Kanervisto, and Rush 2016, the `im2latex100k` dataset has led research forward by removing the overhead of heavy pre-processing. In recent years, it has come under question as a possible bottleneck for progress.

2 Corpus Harvesting

2.1 Challenges

We harvested examples from Cornell's ArXiv archive. As the ArXiv strictly forbids scraping, we found a patron who properly obtained copies and uploaded

them to `archive.org`. We wrote a script to extract the source code files, and notably, we use the `pandoc` document parser.

LaTeX and its subset, TeX pose an unique set of challenges compared to other steps of the pre-processing pipeline. LaTeX is a Turing-Complete language, equivalent in power to general purpose programming languages like C++. It may contain loops and recursion, classifying it as a recursively enumerable language. As such, no LaTeX parser is guaranteed to terminate. From Deng, Kanervisto, and Rush 2016, we also reiterate that several input sequences may map to the same output sequence, so we use the same normalization with Khan Academy's KaTeX called by their scripts.

Practically, `pandoc` handled the files from the corpus gracefully, but due to unlinked files and mismatched, we set `timeout 5` to terminate the program after 5 seconds of stalling. `pandoc` also handles ambiguities to expression matching. Before, plain text not containing mathematical expressions had to be filtered out of the data, but `pandoc` generates an abstract syntax tree and encases the mathematical expressions in the more "specific"[1] expression `\((.*?)\)` compared to the readily-matching `$(.*?)$`. An example of one such false positive that was eliminated is provided in the appendix.

Most importantly, `pandoc` expands all the macros in the source documents. In Deng, Kanervisto, and Rush 2016 the processed formulas required an additional step of filtering out such macros. This is one systemic bias of `im2latex100k`.

3 Topics

4 Deliverables

4.1 Required Objectives

At the submission deadline, we will have the following prepared:

1. All data generated and normalized, building on the work of the Harvard team.
2. All source code containing our finished models and documentation and
3. Research paper outlining the intricacies of our models and their performance on generated examples.
4. Live demo of the model
5. 2 minute video

[1] In that it generates fewer false-positives

4.2 Stretch Goals

Minimal interactive web front-end where a user will be able to upload an image of a mathematical expression and receive the LaTeX code associated with it. This would also serve as the platform for one of our demos.

5 Evaluation

We will evaluate our models with a confusion matrix, Hamming distance, and statistic where appropriate for both - similar to the perplexity metric employed by the Harvard Paper that is a common metric in information theory and machine learning.

References

[DKR16] Yuntian Deng, Anssi Kanervisto, and Alexander M Rush. "What you get is what you see: A visual markup decompiler". In: *arXiv preprint arXiv:1609.04938* 10 (2016), pp. 32–37.