Tree-Based Structure Recognition Evaluation for Math Expressions: Techniques and Case Study

Mahshad Mahdavi
Document and Pattern Recognition Lab
Rochester Institute of Technology
Rochester, NY, USA
mxm7832@rit.edu

Richard Zanibbi
Document and Pattern Recognition Lab
Rochester Institute of Technology
Rochester, NY, USA
rxzvcs@rit.edu

I. Introduction

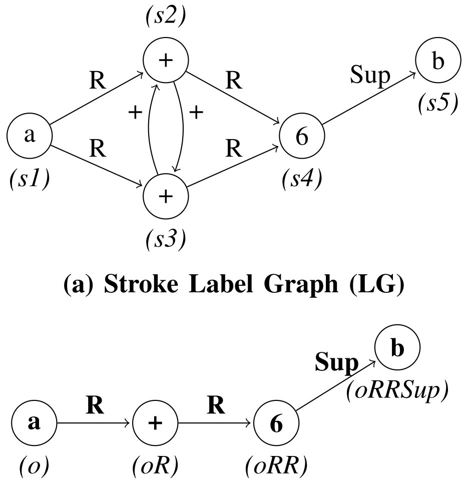
Evaluating visual structure recognition for math expressions is complex due to the interactions between input primitives, detected symbols, and their spatial relationships. Visual structure is expressed using trees describing the arrangement of symbols on writing lines, and the hierarchical spatial arrangement of these writing lines (see Figure 1(b)). LaTeX formulas represent this information along with additional formatting directives (e.g., for fonts, symbol sizes, and spacing).

Formula recognition comprises three major tasks: detecting symbols, classifying symbols, and determining spatial relationships between symbols. With the correct tools, symbols and relationships can be evaluated separately, and specific errors compiled using confusion matrices and *confusion histograms* that tabulate and count specific errors for given sub-trees in ground truth formulas [1]. As a simple illustration, if the expression '$xy+1$' is recognized as '$2a+b$,' the output can be considered as having the correct spatial relationships/structure (i.e., five symbols on one writing line), but only one symbol is shared between the two formulas ('+').

A number of state-of-the-art systems were inspired by automated image captioning, avoiding explicit symbol segmentation and producing LaTeX output [2], [3]. By representing structure only over detected symbol classes, we lose the correspondence of input primitives (e.g., handwritten strokes in Figure 1(a)) to symbols in the output. So far, LaTeX outputs have been evaluated using string-based metrics such as exact matching, string edit distances, or n-gram-based metrics such as BLEU, or by computing distances between images produced after rendering TeX formulas. Unfortunately, different LaTeX strings can represent identical formulas, or produce differences in spacing or formatting for the same underlying expression. These measures approximate rather than directly capture differences in visual structure at the level of symbols and relationships seen in Figure 1(b).

II. Visual Structure Representation (symLG) and Symbol-Level Evaluation Metrics

To allow us to evaluate and compare visual structure recognition using symbols and relationships directly, we convert LaTeX and other structure representations to Symbol Label Graphs (*symLGs*, see Figure 1(b)). In a symLG, each symbol has an identifier. Identifiers are defined by the sequence of spatial relationships from the root symbol to each symbol in the tree. The adjacency matrix for a symLG contains symbol class labels on the diagonal (for symbol self-edges), and spatial relationships between symbols in off-diagonal entries. Using this, we can then directly compare formulas based on the agreement between adjacency matrix entries, even when symbol identifiers are missing in one of the graphs [1].



(a) Stroke Label Graph (LG)

(b) Symbol Label Graph (symLG)

Fig. 1: Different representations for formula '$a + 6^b$' written using five strokes. Node identifiers are shown in brackets.

To obtain symLGs, we first need to produce a uniform symbol-level structure representation, for which we have used Presentation MathML. We convert TeX formulas to MathML using `pandoc`.[1] This transformation preserves symbols and spatial relationships while removing formatting directives (e.g., `\quad`, fonts). For primitive-based output representations, such as the stroke label graph in Figure 1(a) [4], we use a simple transducer to produce MathML. In a label graph, all strokes in a symbol share the same spatial relationship with strokes in the related symbol (e.g., for 'a' and '+'), and symbol segmentation is given using bidirectional edges labeled with

[1] https://pandoc.org

TABLE I: symLG im2latex-100k results (9,378 test formulas). Shown are correct symbol/relationship locations (Detection), symbol/relationship classes (Det.+Class), formula SLT structure ignoring symbol labels, and valid structure and symbol labels (Str.+Class).

	Symbols		**Relationships**		**Formulas**	
	Det.	Det.+Class	Det.	Det.+Class	Str.	Str.+Class
IM2TEX	95.70	93.48	95.50	95.50	86.79	**83.15**

their associated symbol's class (e.g., for '+').

Once we have a MathML representation for a formula, we generate symbol identifiers using the spatial relationship sequence from the root symbol (see Figure 1(b)). Identifiers allow us to address symbols on writing lines from different structure representations. This produces a *symbolic* representation for recognition outputs, one that ignores the correspondence of output symbols to input data [1].

Symbol-Level Metrics. Once we have our symLG representation, we compute symbol-level metrics using evaluation tools from the CROHME handwritten math recognition competitions [1], [4], [5] originally designed for stroke-level evaluation (the LgEval library). LgEval metrics include formula and symbol recognition rates, along with recall and precision for detection and detection + classification of both symbols and relationships [1]. The symLG representation allows us to identify specific relationship classification errors, structure errors, and symbol classification errors (*when symbol locations/identifiers are correct*; see Section IV).

Related Work. Symbolic evaluation has been considered previously, e.g., EMERS [6] is a tree edit distance using an Euler string representation to quantify partially correct recognition for MathML trees. Symbol errors are weighted inversely proportional to their distance from the main writing line (baseline) of the expression, to decrease the impact of errors inside branches. A form of symbolic evaluation based on unlabeled trees was used in early CROHME competitions [1]. The IMEGE metric [7] is a pixel-based image distance metric, which has been used for evaluation by rendering an image from output LaTeXstrings [2].

III. Case Study

We use symLGs to evaluate the IM2TEX system by Deng et al. [2]. As shown in Table 1, our symLG metrics provide measures for correct symbol detection (i.e., symbols exist at expected spatial locations), correct symbol locations and labels, correct relationships, and structure and symbol classification accuracy at the expression level. Note that because spatial relationships determine symbol locations, a correctly *detected* relationship is also correctly classified.

For the im2latex-100k data set, we were able to convert 9,378 of the 10,355 test formulas (90.6%) from LaTeX to MathML using `pandoc`. Many failed conversions are caused by invalid syntax (e.g., missing brackets).

For the 9,378 formulas that were converted successfully to MathML, We are now able to report that the percentage of correct formulas with both correct symbols and structure is 83.15%, that 93.48% of symbols are in the proper location with their correct class, and that 95.50% of spatial relationships are correct. The metrics previously reported by the IM2TEX authors include BLEU (tok) at 58.41, BLEU (norm) at 87.73, exact image-based pixel matching of 77.46, and image-based pixel matching with a whitespace tolerance (-ws) of 79.88 [2].

Moreover, using symLGs we can provide detailed error analysis that string and image-based representations cannot capture (omitted for space). The most common error is 'missing' symbols. This happens because symbols are identified by their absolute path - therefore, errors in structure lead to errors in symbol detection and classification. Note that this also means that correctly detected symbols at the incorrect position in a symLG are identified as invalid.

IV. Conclusion

We have presented a technique that allows string and tree-based formula structure representations to be meaningfully compared at the level of recognized symbols and relationships. Further, this permits fine-grained evaluation of recognition results at the individual symbol and relationship level, as well as at the expression level, addressing limitations with the previous use of string-based and image-based metrics used to evaluate LaTeX output. In future work, we hope to use more robust methods for converting from LaTeX to MathML.

Our symLG-based metrics were used for the recent ICDAR 2019 CROHME + TFD competition [8], as they are simple to understand, and provide useful global performance metrics and automated error analyses.

References

[1] H. Mouchère, R. Zanibbi, U. Garain, and C. Viard-Gaudin, "Advancing the state of the art for handwritten math recognition: the CROHME competitions, 2011–2014," *Int'l. J. Document Analysis and Recognition*, vol. 19, no. 2, pp. 173–189, 2016.

[2] Y. Deng, A. Kanervisto, J. Ling, and A. M. Rush, "Image-to-markup generation with coarse-to-fine attention," *arXiv preprint arXiv:1609.04938*, 2016.

[3] J. Zhang, J. Du, S. Zhang, D. Liu, Y. Hu, J. Hu, S. Wei, and L. Dai, "Watch, attend and parse: An end-to-end neural network based approach to handwritten mathematical expression recognition," *Pattern Recognition*, vol. 71, pp. 196–206, 2017.

[4] R. Zanibbi, A. Pillay, H. Mouchere, C. Viard-Gaudin, and D. Blostein, "Stroke-based performance metrics for handwritten mathematical expressions," in *2011 International Conference on Document Analysis and Recognition*. IEEE, 2011, pp. 334–338.

[5] R. Zanibbi, H. Mouchere, and C. Viard-Gaudin, "Evaluating structural pattern recognition for handwritten math via primitive label graphs," in *Document Recognition and Retrieval XX*, vol. 8658. International Society for Optics and Photonics, 2013, p. 865817.

[6] K. Sain, A. Dasgupta, and U. Garain, "Emers: a tree matching–based performance evaluation of mathematical expression recognition systems," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 14, no. 1, pp. 75–85, 2011.

[7] F. Álvaro, J.-A. Sánchez, and J.-M. Benedí, "An image-based measure for evaluation of mathematical expression recognition," in *Iberian Conference on Pattern Recognition and Image Analysis*. Springer, 2013, pp. 682–690.

[8] M. Mahdavi, R. Zanibbi, H. Mouchère, and U. Garain, "ICDAR 2019 CROHME + TFD: Competition on Recognition of Handwritten Mathematical Expressions and Typeset Formula Detection," in *Proc. ICDAR 2019*, to appear.