A Math Formula Extraction and Evaluation Framework for PDF Documents

Ayush Kumar Shah, Rochester Institute of Technology, USA

Abhisek Dey, Rochester Institute of Technology, USA

Richard Zanibbi, Rochester Institute of Technology, USA

{as1211, ad4529, rxzvcs}@rit.edu

Document and Pattern Recognition Lab

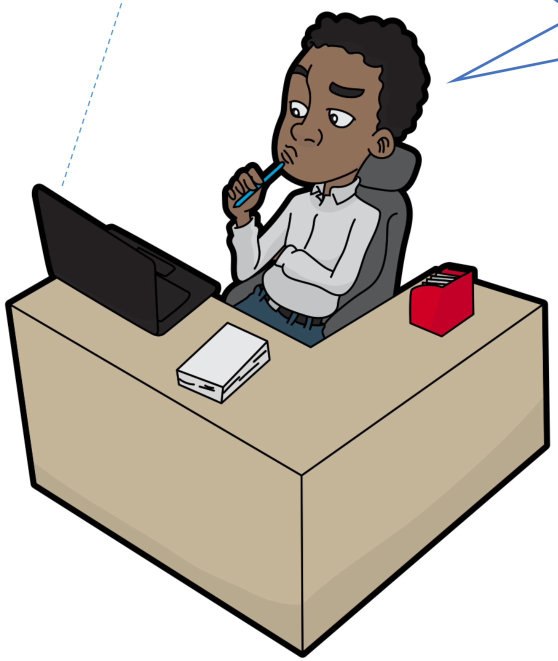
RIT Rochester Institute of Technology

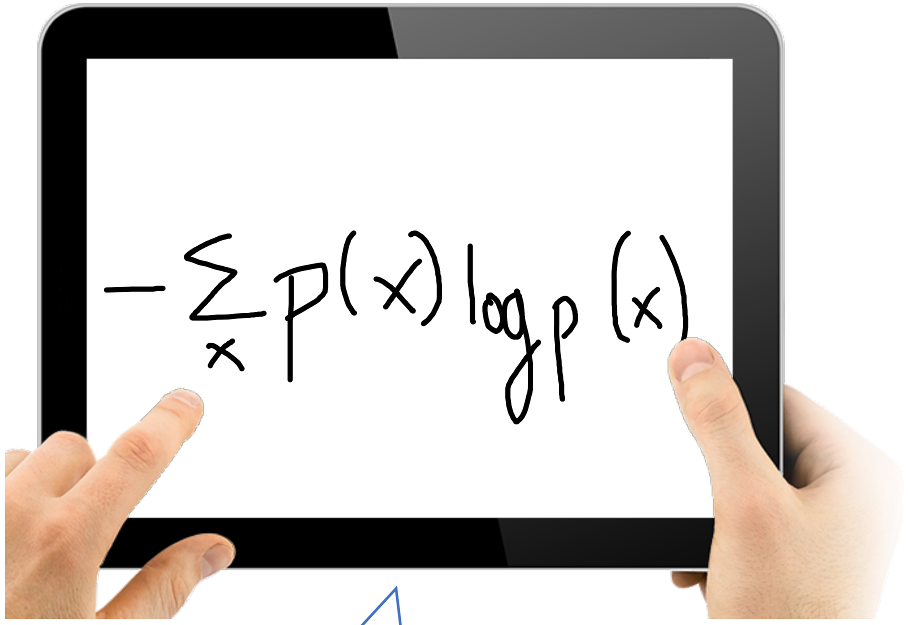
ICDAR LAUSANNE 2021

Motivation: Information Retrieval

$$-\sum_x p(x) \log p(x)$$

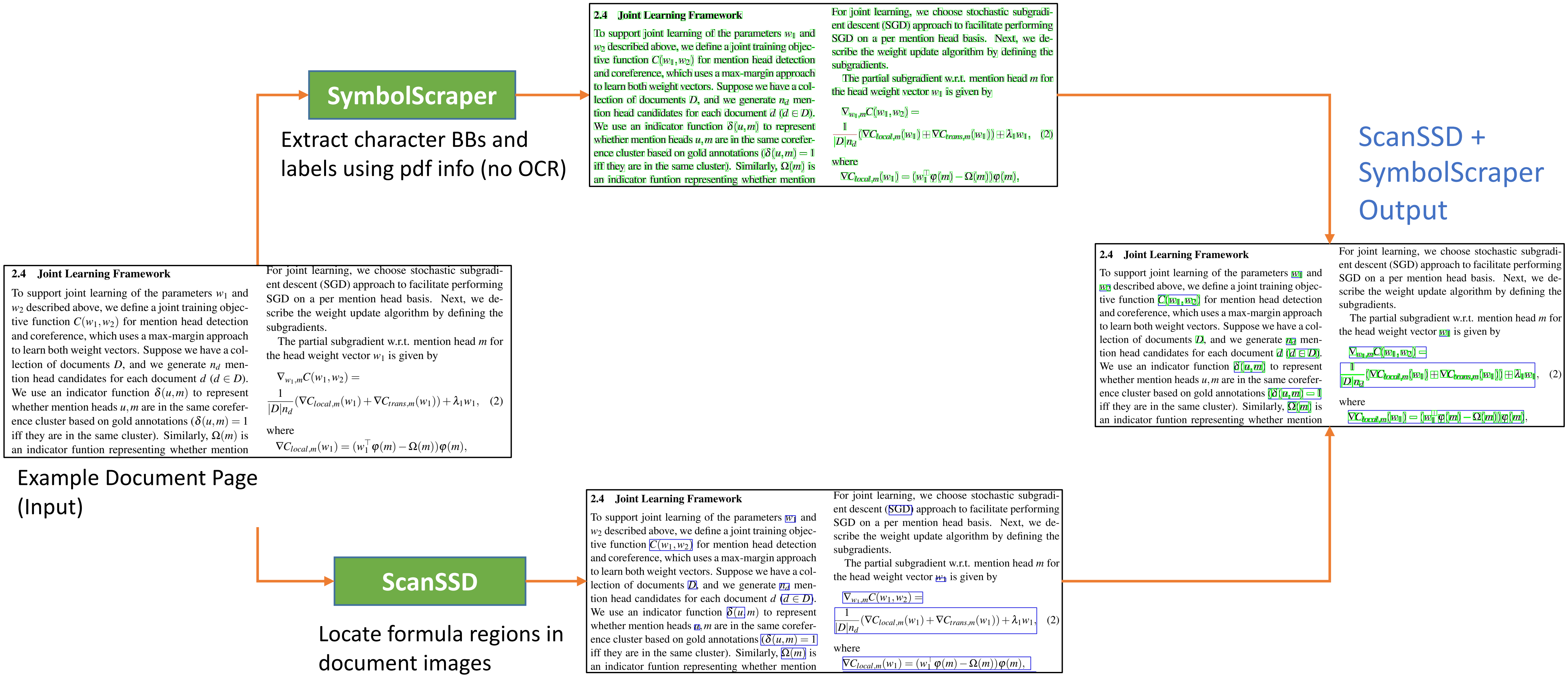
What is the name
of this formula?



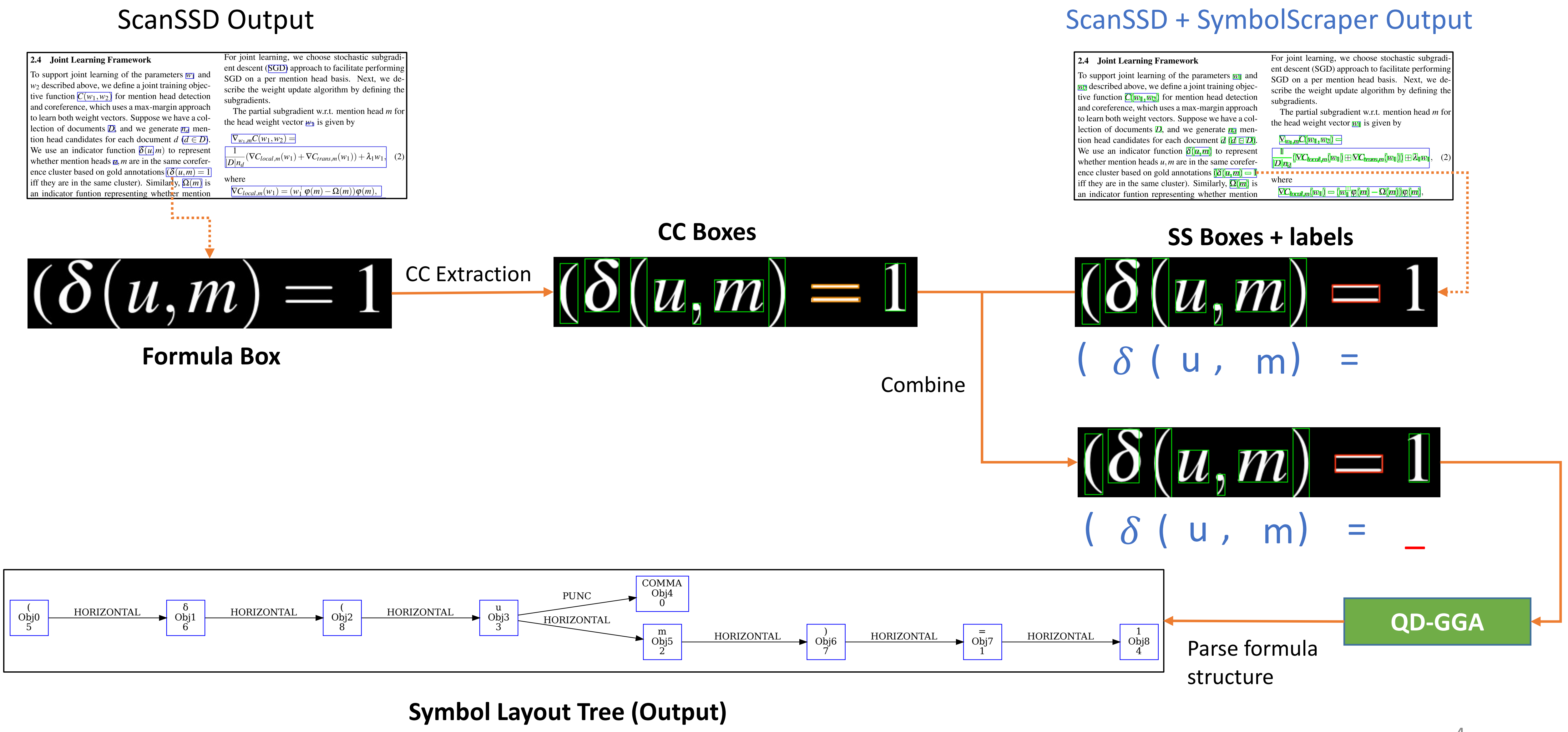


Let's find documents
describing this formula.

Mathematical Formula Extraction: Overview

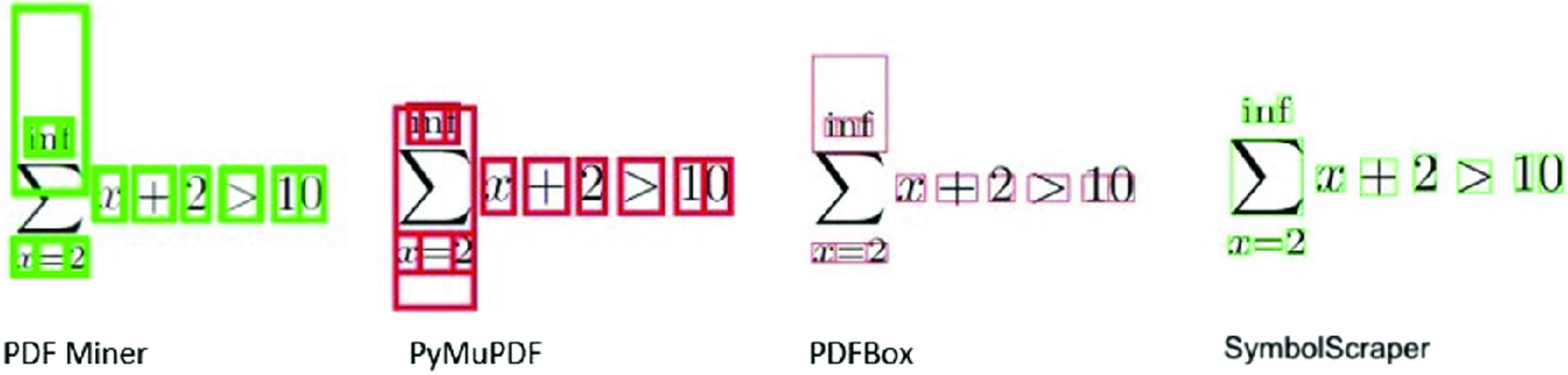


Mathematical Formula Extraction: Overview



SymbolScraper: Extracting Symbols in PDF

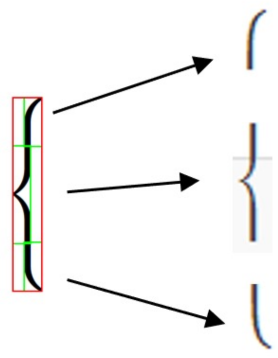
- Based on Apache PDFBox
- Avoids OCR in **born-digital** PDF documents and instead uses vector drawing commands in PDF
- Unicode, writing line position and attributes derived from PDF encoding
- *‘em box’* or underlying character outlines (glyphs) represent symbol outlines in a font as boxes



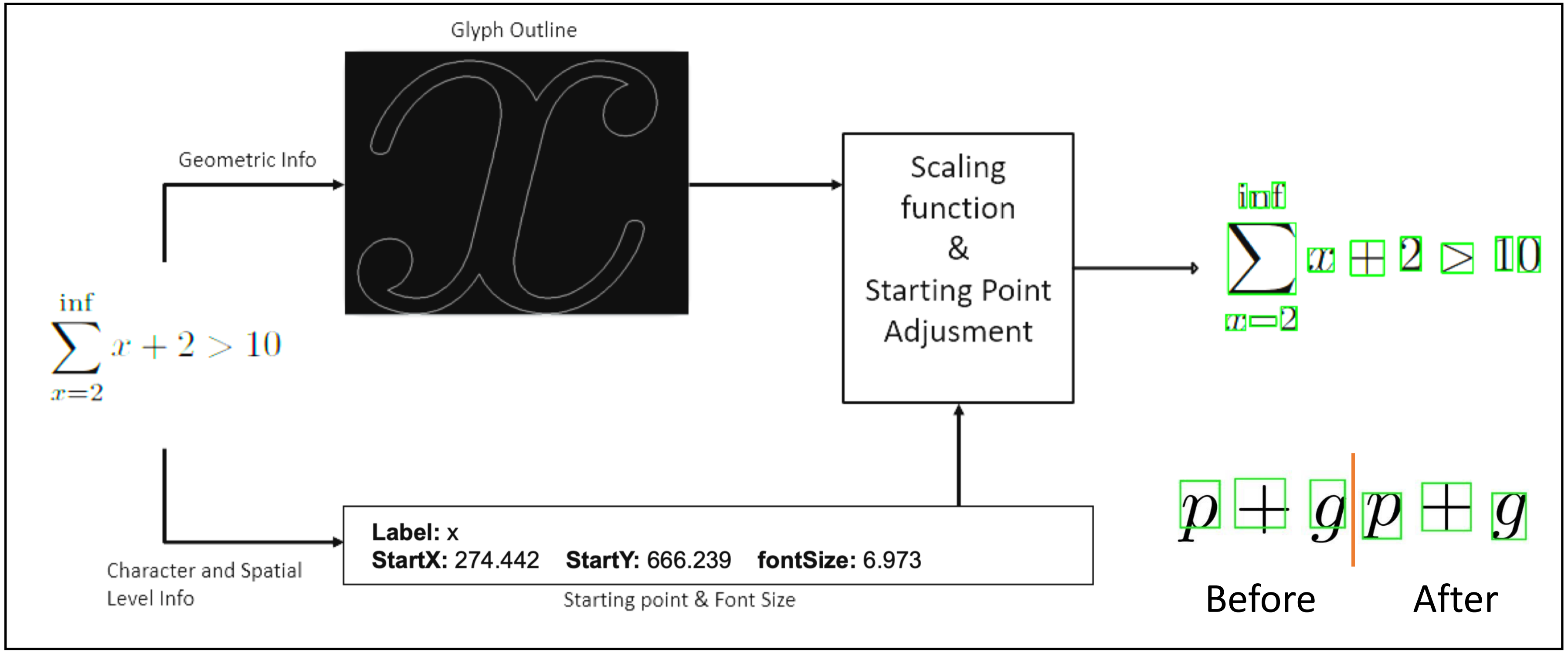
Unlike other methods, SymbolScraper uses glyphs to fine-tune bounding box locations

SymbolScraper: Extracting Symbols in PDF

- Glyphs and font scaling information used to obtain precise bounding box locations
- Compound characters (large braces, square roots, etc.) are formed of 2 or more characters



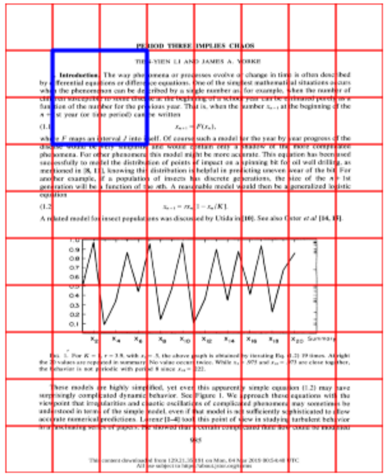
Compound Symbol



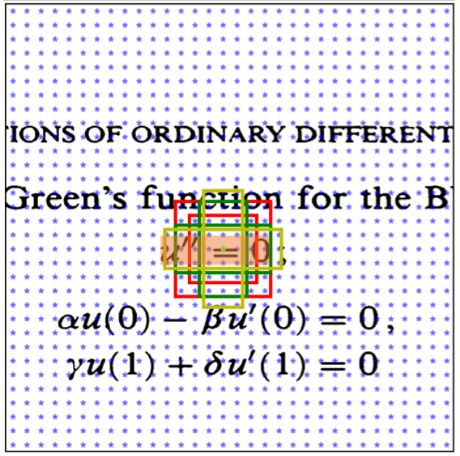
Correcting glyph origins

ScanSSD: Locating Formula Regions

- **Scanning Single-Shot Detector,** CNN which locates formula bounding boxes using a sliding window
- 600 dpi images broken into windows of 1200 x 1200 pixels, SSD applied in each window at 10% stride
- Non-Maximal Suppression selects the highest confidence regions from overlapping detections
- Wider default boxes sizes used with aspect ratios of 5, 7, and 10 -> increased recall



A window (in Blue) slides across the grid (in Red) (50% stride)

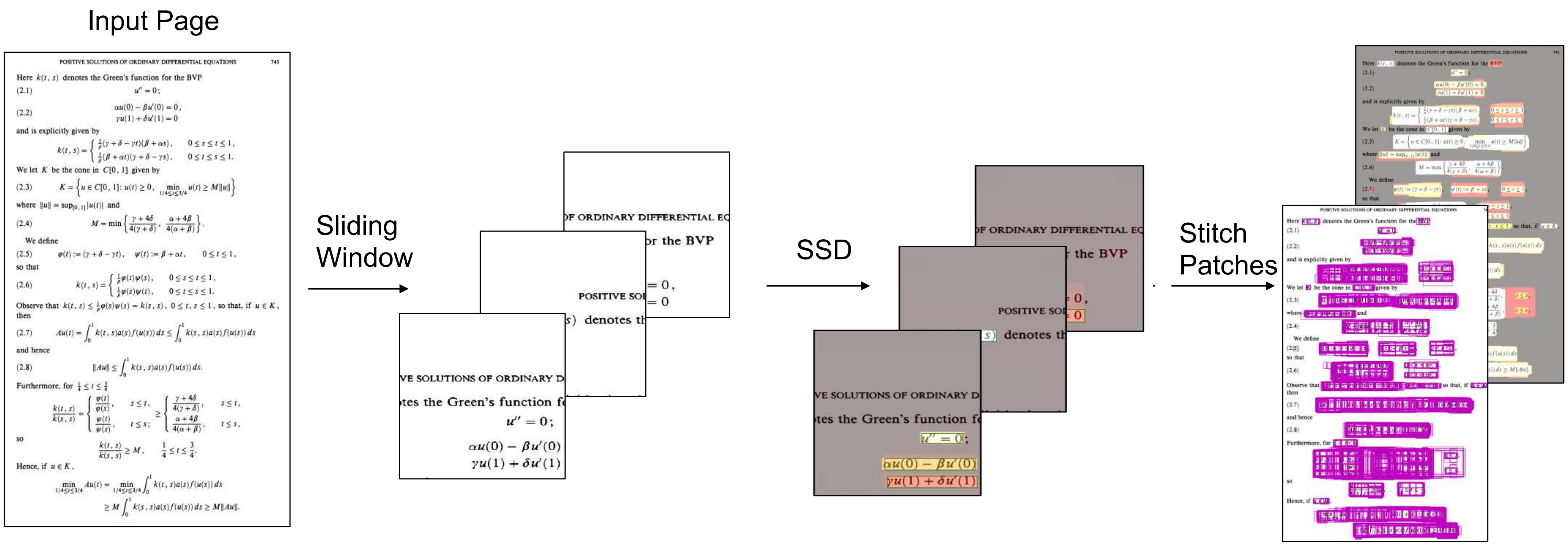


Default boxes around a grid point

- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: **SSD: Single shot multibox detector**. In: European conference on computer vision. pp. 21–37. Springer (2016)
- Mali, P., Kukkadapu, P., Mahdavi, M., Zanibbi, R.: **ScanSSD: Scanning Single Shot Detector for Mathematical Formulas in PDF Document Images**. arXiv:2003.08005 [cs] (2020)

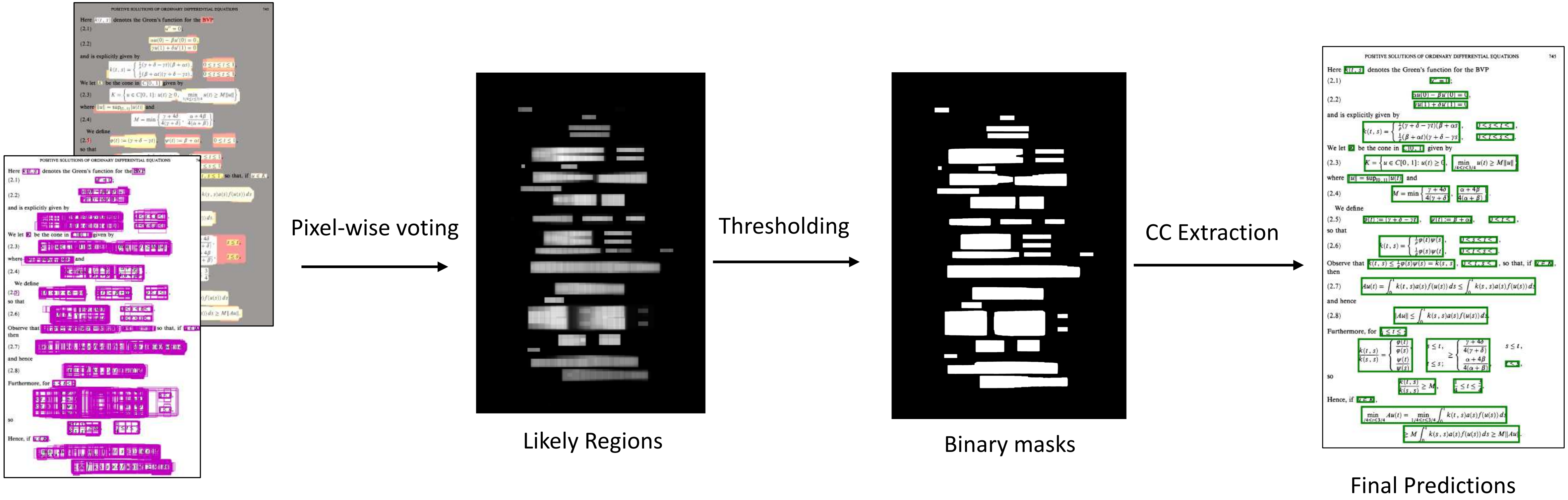
ScanSSD: Locating Formula Regions

- A sliding window divides the page into windows which are processed by ScanSSD
- The partial predictions at the window-level are pooled together and the final regions are identified using pixel-wise voting (stitching)



- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: **SSD: Single shot multibox detector**. In: European conference on computer vision. pp. 21–37. Springer (2016)
- Mali, P., Kukkadapu, P., Mahdavi, M., Zanibbi, R.: **ScanSSD: Scanning Single Shot Detector for Mathematical Formulas in PDF Document Images**. arXiv:2003.08005 [cs] (2020)

ScanSSD: Locating Formula Regions

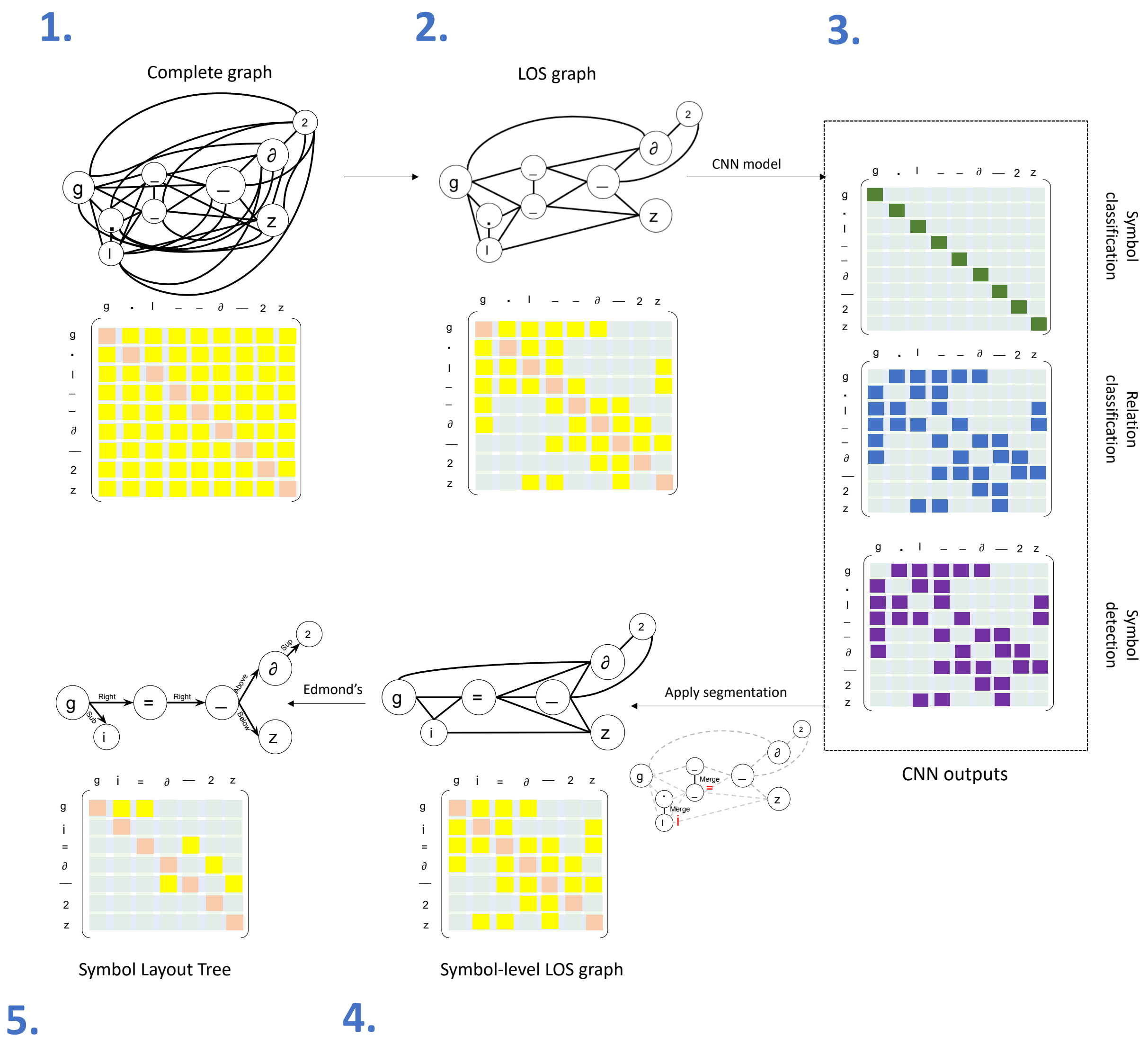


Bottom: Window-level Predictions
Top: Confidence masks

- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: **SSD: Single shot multibox detector**. In: European conference on computer vision. pp. 21–37. Springer (2016)
- Mali, P., Kukkadapu, P., Mahdavi, M., Zanibbi, R.: **ScanSSD: Scanning Single Shot Detector for Mathematical Formulas in PDF Document Images**. arXiv:2003.08005 [cs] (2020)

QD-GGA: Recognizing Formula Structure (Parsing)

1. **Construct** graph over CCs
2. **Prune:** Convert to LOS graph
3. **Classify** edges as merge/split and relationships, nodes as symbols
4. **New LOS graph:** detected symbols
5. **Extract MST** using Edmond's arborescence algorithm

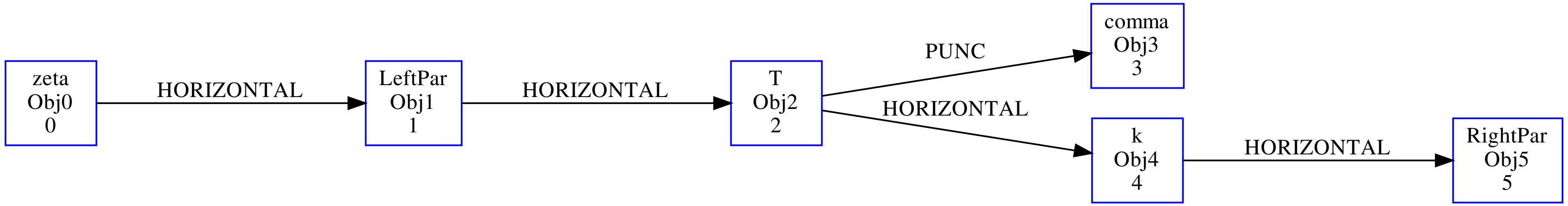


Mahdavi, M.; Sun, L.; Zanibbi, R.: **Visual Parsing with Query-Driven Global Graph Attention (QD-GGA)**. In Conference on Computer Vision and Pattern Recognition Workshops (2020)

Inputs

Input Formula with
CCs or Symbols

Outputs



Symbol Layout Tree (SLT)

```
<math xmlns="http://www.w3.org/1998/Math/MathML">
<mrow>
    <mi xml:id="0:">ζ</mi>
    <mrow>
        <mo xml:id="1:">(</mo>
        <mrow>
            <mrow>
                <mi xml:id="2:">T</mi>
                <mo xml:id="3:">,</mo>
            </mrow>
            <mrow>
                <mi xml:id="4:">k</mi>
                <mo xml:id="5:">)</mo>
            </mrow>
        </mrow>
    </mrow>
</mrow>
</math>
```

SLT in MathML

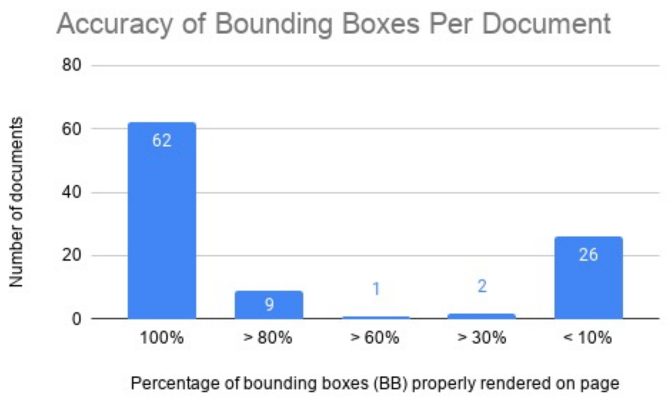
```
\(\zeta\left( {{T,}\left. k \right)} \right.\)
```

SLT in LATEX

Mahdavi, M.; Sun, L.; Zanibbi, R.: **Visual Parsing with Query-Driven Global Graph Attention (QD-GGA)**. In Conference on Computer Vision and Pattern Recognition Workshops (2020)

SymbolScraper Results

Summary of SymbolScraper Accuracy



Hardware Specifications and Speed	
Storage	HDD
Dataset Size	100 pages
Total time	28 mins, 19 secs
Average time	1.7 secs/page

ScanSSD Results

Formula Detection Results for TFD-ICDAR2019

	IOU $\geq$ 0.75			IOU $\geq$ 0.5		
	Precision	Recall	F-score	Precision	Recall	F-score
ScanSSD	**0.774**	**0.690**	**0.730**	**0.851**	**0.759**	**0.802**
RIT 2	0.753	0.625	0.683	0.831	0.670	0.754
RIT 1	0.632	0.582	0.606	0.744	0.685	0.713
Mitchiking	0.191	0.139	0.161	0.369	0.270	0.312
Samsung*	0.941	0.927	0.934	0.944	0.929	0.936

*Used character information

Hardware Specifications and Speed	
Storage	HDD
RAM	32 GB
Graphics	Nvidia RTX 2080 Ti
Processor	AMD Ryzen 7 2700
Dataset Size	233 pages
Total time	4 hrs, 33 mins, 31 secs
Average time	70.4 secs/page

QD-GGA Results

Formula Recognition Results for InftyMCCDB-2[1] Test set

Metrics	Value
Structure rate	92.56
Structure + Classification rate	85.94

Hardware Specifications and Speed	
Storage	HDD
RAM	32 GB
Graphics	Nvidia GTX 1080
Processor	Intel(R) Core(TM) i7-9700KF
Dataset Size	6830 images
Total time	26 mins, 25 secs
Average time	232 ms/formula

- [1] https://zenodo.org/record/3483048#.XaCwmOdKjVo
- Mali,P.,Kukkadapu,P.,Mahdavi,M.,Zanibbi,R.:**ScanSSD:ScanningSingleShot Detector for Mathematical Formulas in PDF Document Images**. arXiv:2003.08005 [cs] (2020)
- Mahdavi, M.; Sun, L.; Zanibbi, R.: **Visual Parsing with Query-Driven Global Graph Attention (QD-GGA)**. In Conference on Computer Vision and Pattern Recognition Workshops (2020)

Recognition Results Visualization (HTML)

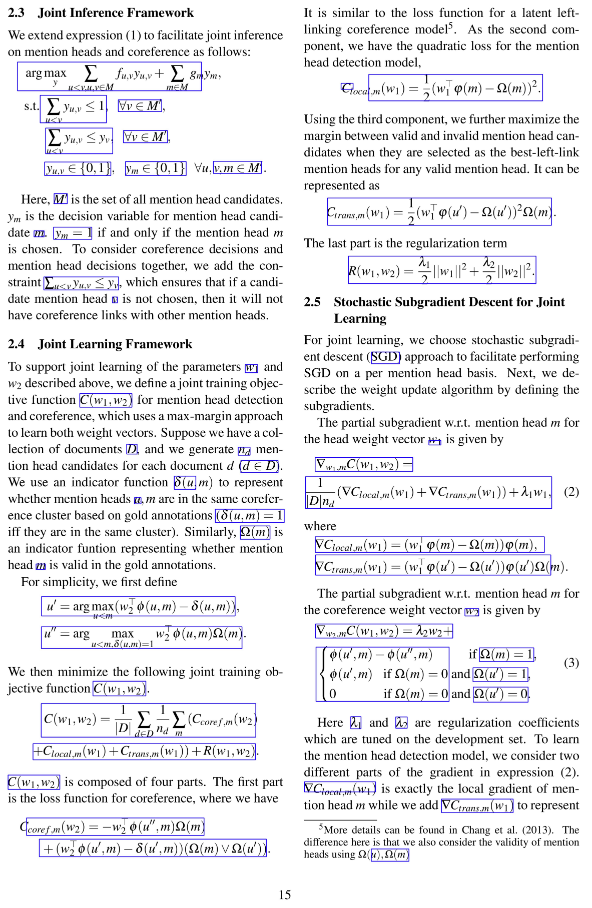
MathSeer Pipeline Results Visualization

Pdf name: K15-1002

Page: 4

Previous page | Home | Next page

Page image



2.3 Joint Inference Framework

We extend expression (1) to facilitate joint inference on mention heads and coreference as follows:

Here, M is the set of all mention head candidates. y_m is the decision variable for mention head candidate m. $y_m = 1$ if and only if the mention head m is chosen. To consider coreference decisions and mention head decisions together, we add the constraint $\sum_{u<v} y_{u,v} \leq y_v$, which ensures that if a candidate mention head v is not chosen, then it will not have coreference links with other mention heads.

2.4 Joint Learning Framework

To support joint learning of the parameters w_1 and w_2 described above, we define a joint training objective function $C(w_1, w_2)$ for mention head detection and coreference, which uses a max-margin approach to learn both weight vectors. Suppose we have a collection of documents D, and we generate n_d mention head candidates for each document d ($d \in D$). We use an indicator function $\delta(u, m)$ to represent whether mention heads u, m are in the same coreference cluster based on gold annotations ($\delta(u, m) = 1$ iff they are in the same cluster). Similarly, $\Omega(m)$ is an indicator funtion representing whether mention head m is valid in the gold annotations.

For simplicity, we first define

We then minimize the following joint training objective function $C(w_1, w_2)$.

$C(w_1, w_2)$ is composed of four parts. The first part is the loss function for coreference, where we have

It is similar to the loss function for a latent left-linking coreference model[5]. As the second component, we have the quadratic loss for the mention head detection model,

Using the third component, we further maximize the margin between valid and invalid mention head candidates when they are selected as the best-left-link mention head for any valid mention head. It can be represented as

The last part is the regularization term

2.5 Stochastic Subgradient Descent for Joint Learning

For joint learning, we choose stochastic subgradient descent (SGD) approach to facilitate performing SGD on a per mention head basis. Next, we describe the weight update algorithm by defining the subgradients.

The partial subgradient w.r.t. mention head m for the head weight vector w_1 is given by

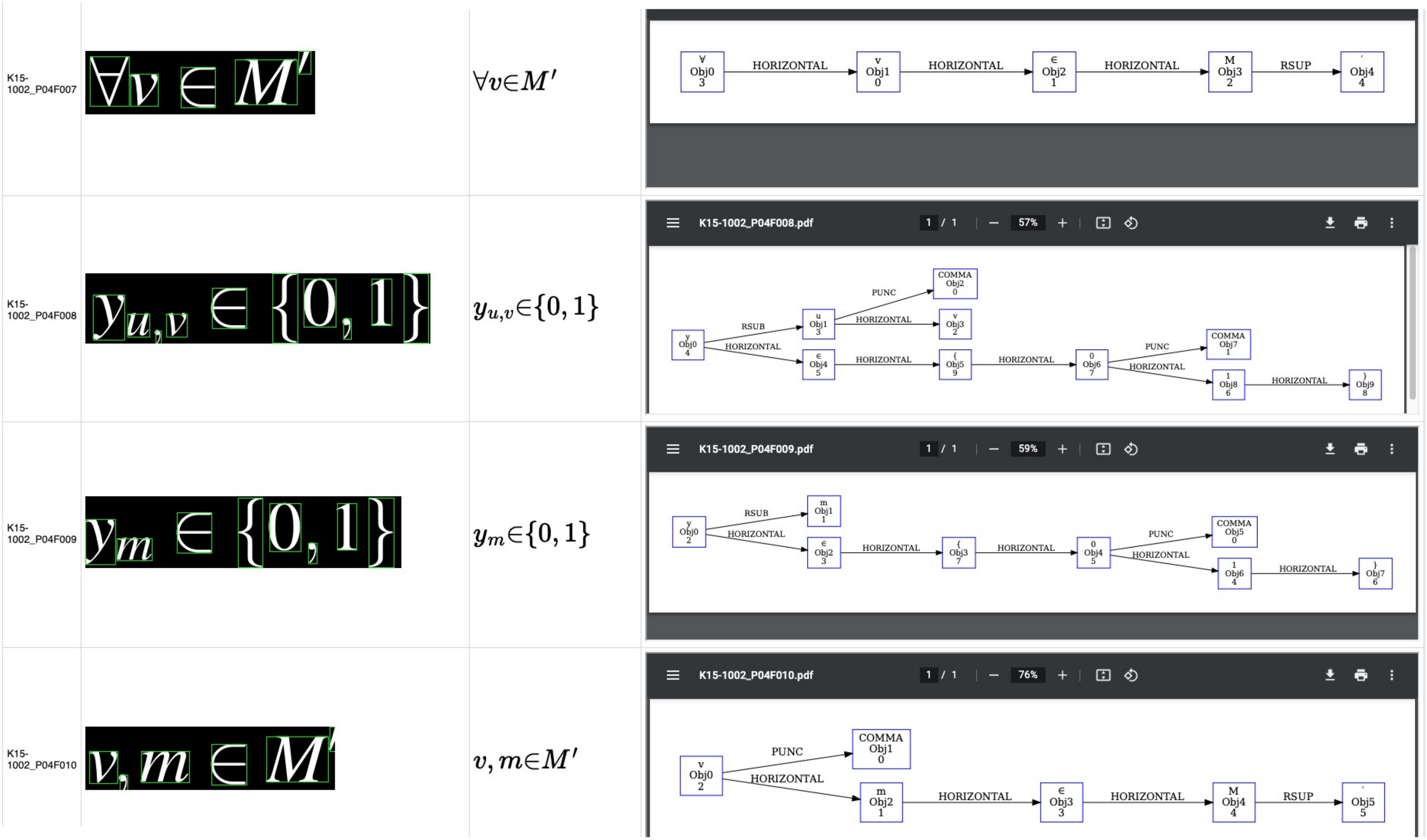
where

The partial subgradient w.r.t. mention head m for the coreference weight vector w_2 is given by

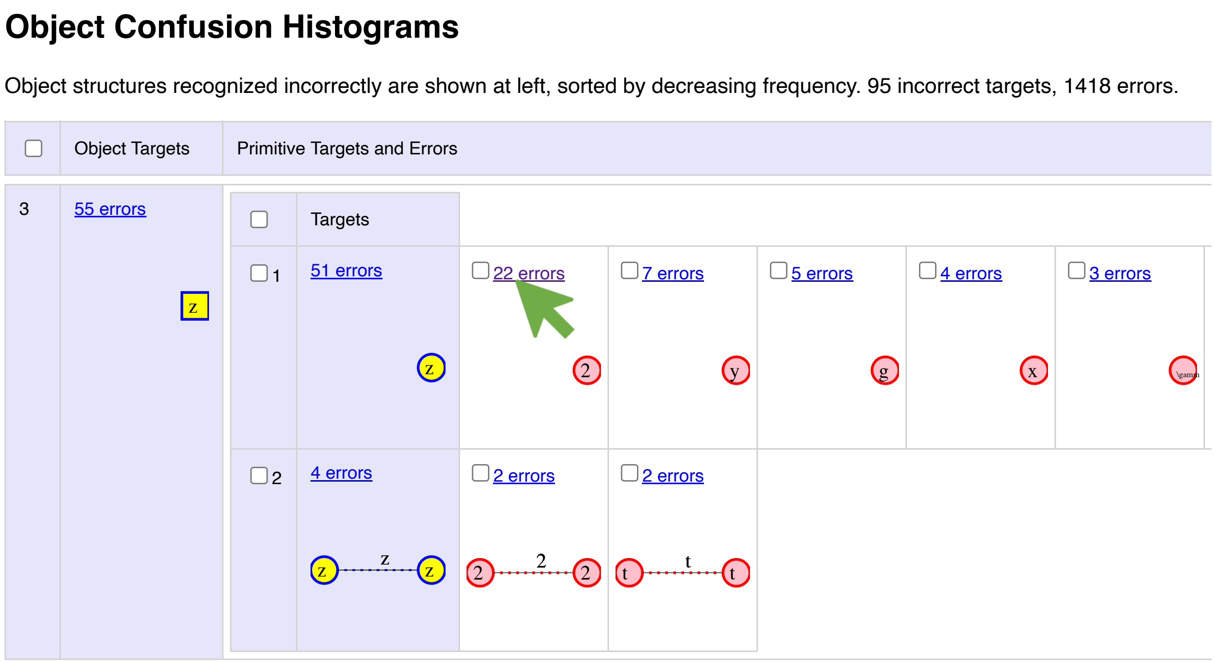
Here λ_1 and λ_2 are regularization coefficients which are tuned on the development set. To learn the mention head detection model, we consider two different parts of the gradient in expression (2). $\nabla C_{local,m}(w_1)$ is exactly the local gradient of mention head m while we add $\nabla C_{trans,m}(w_1)$ to represent

[5]More details can be found in Chang et al. (2013). The difference here is that we also consider the validity of mention heads using $\Omega(u), \Omega(m)$

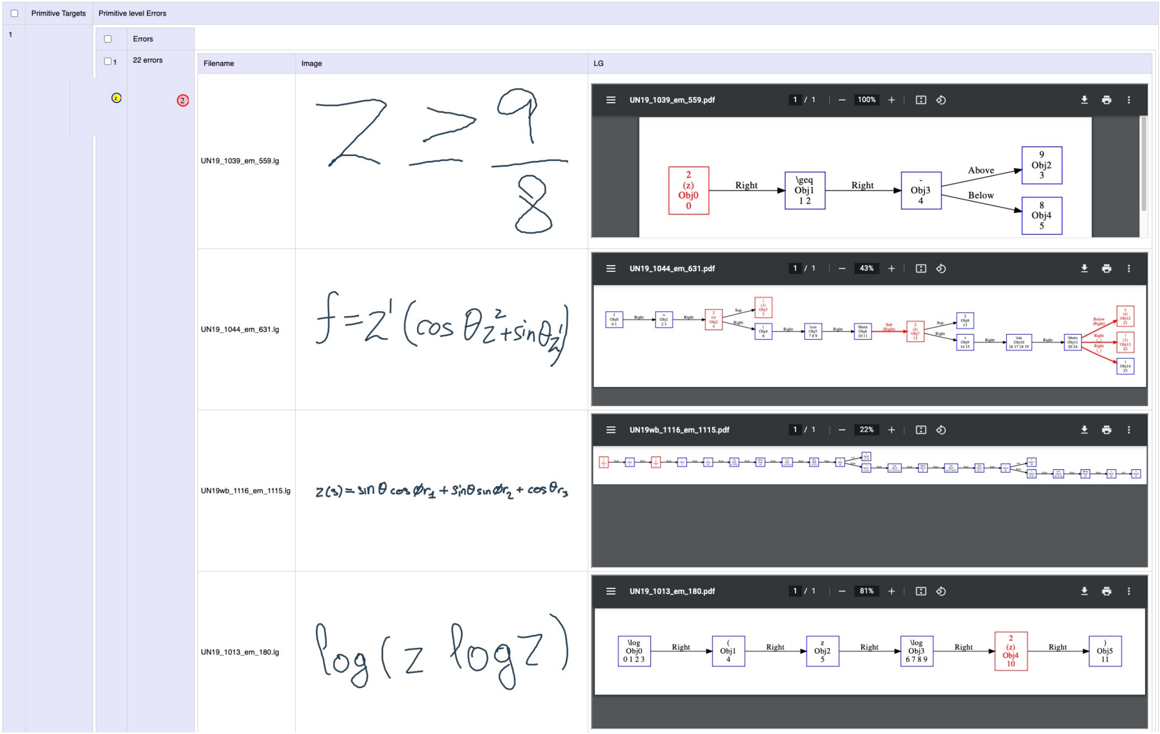
15



LgEval Extension: Error Visualization

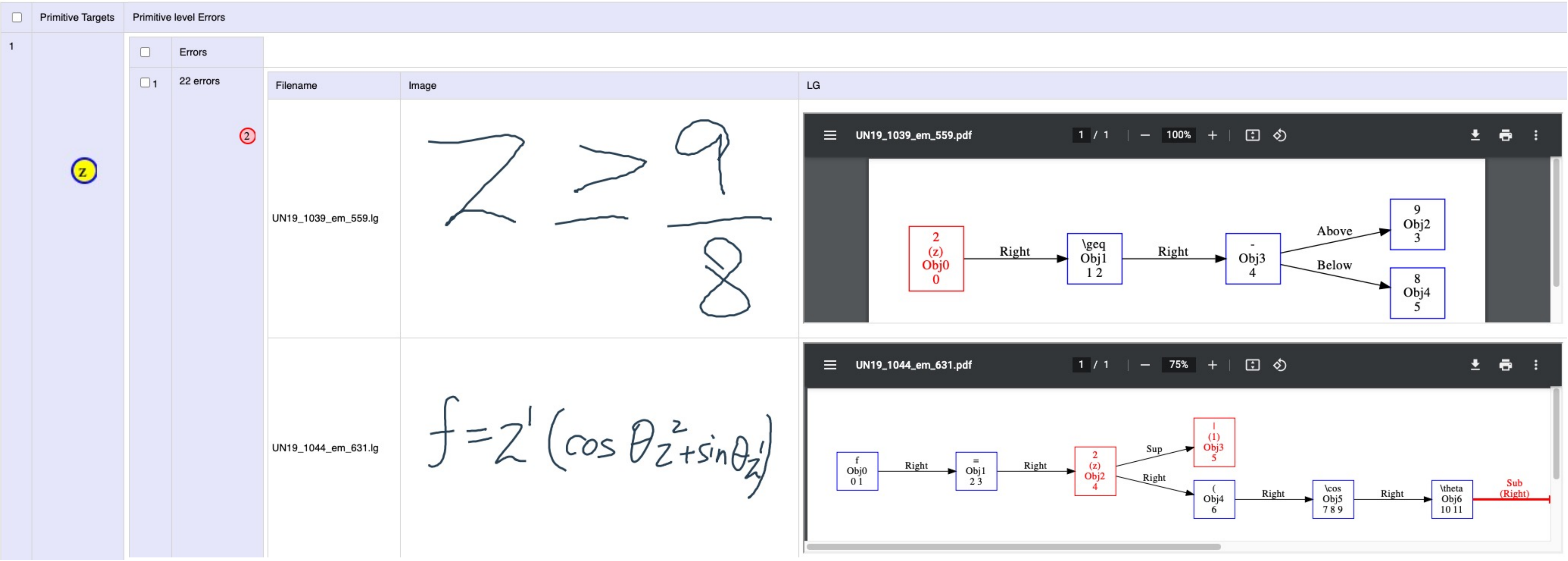


Errors organized by decreasing frequency



Specific instances where 'z' is misclassified as '2,' seen after clicking on the '22 errors' link

LgEval Extension: Error Visualization



Zoomed in: Specific instances where ‘z’ is misclassified as ‘2,’ seen after clicking on the ‘22 errors’ link

Conclusion and Future Work

- Open-source formula extraction pipeline for PDF documents
  - https://www.cs.rit.edu/~dprl/software.html
- PDF symbol extractor that identifies precise bounding box locations in born-digital PDFs
- A simple and effective algorithm for detection of math expressions using visual features alone
- Extended tools for visualizing recognition results and formula parsing errors
- **ScanSSD-XYc:** Unified page and window level merging using recursive XY Cuts avoiding NMS speeding up detection by 300 times approximately (included in the repository)

Future work

- **SymbolScraper:** Handle Type 3 Fonts and faster system for symbol extraction, better handling of compound characters
- **Pipeline:** End-to-End trainable system for detection and parsing

- Dey, A.; Zanibbi, R.: ScanSSD-XYc: Faster Detection of Math Formulas. In The 14th IAPR International Workshop on Graphics Recognition (GREC 2021), to appear; 2021.

Thank You

This material is based on upon work supported by the Alfred P. Sloan Foundation under Grant No. G-2017-9827 and the National Science Foundation (USA) under Grant Nos. IIS-1717997 (MathSeer project) and 2019897 (MMLI project)

Thanks to R. Joshi, P. Mali, P. Kukkadapu, A. Keller, M. Mahdavi, and J. Diehl for their contribution to the formula extraction pipeline. Jian Wu provided the document collected used to evaluate SymbolScraper

NSF

Alfred P. Sloan FOUNDATION

dprl Document and Pattern Recognition Lab