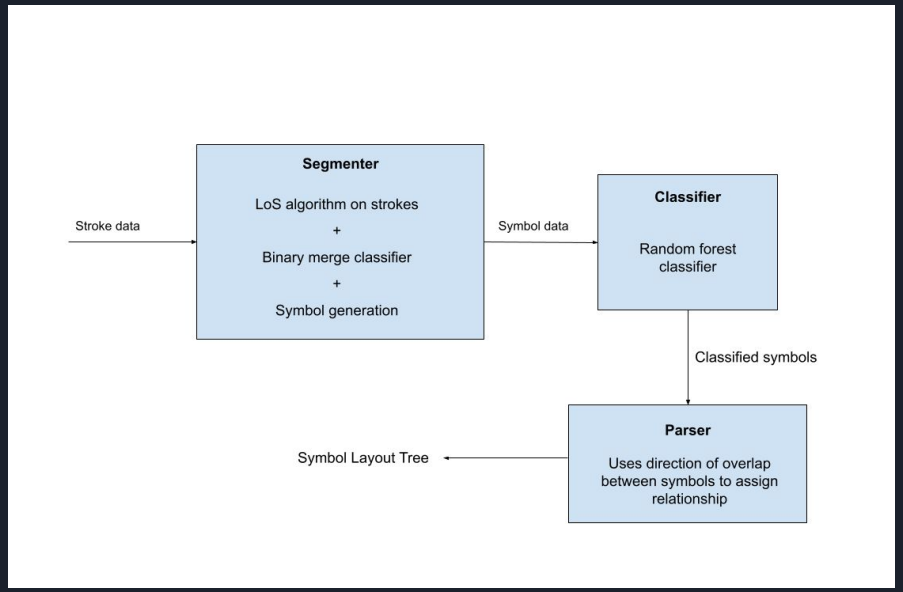
Parsing Handwritten Math Expressions using Geometric Overlap between Symbols

Anushree Das
Gautam Gadipudi
Nishi Parameshwara
Raj Bhensadadia

Approach Used

- Segmentation [1]
  - Line of Sight (LoS) graph generation on stroke data
  - Use a binary classifier to assign edge labels as '*' or '_'
  - Group strokes with edge labels as '*' using Algo 2 [1]
- Classification
  - Normalize and resample the symbol data [3]
  - Extract features [2]
  - Classify the symbols
- Parsing
  - Construct a complete graph using symbols as the nodes and minimum distance between them as the edges.
  - Extract a Minimum Spanning Tree (MST) from the complete graph
  - Assign relationship based on the direction of overlap among symbols.



[1] L. Hu and R. Zanibbi. "Line-of-Sight Stroke Graphs and Parzen Shape Context Features for Handwritten Math Formula Representation and Symbol Segmentation". In:2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR). 2016, pp. 180–186.doi:10.1109/ICFHR.2016.0044

[2] Lei Hu and Richard Zanibbi. "HMM-based recognition of online handwritten mathematical symbols using segmental k-means initialization and a modified pen-up/down feature". In:2011 International Conference on Document Analysis and Recognition. IEEE. 2011, pp. 457–462.

[3] Mois´es Pastor, Alejandro Toselli, and Enrique Vidal. "Writing speed normalization for on-line handwritten text recognition". In:Eighth International Conference on Document Analysis and Recognition (ICDAR'05).IEEE. 2005, pp. 1131–1135

Features [1]

1. **Total Length**
   Summation of total length of all the traces.

2. **Vicinity Slope α**
   Gives model an idea of the slope that is observed at the points present in the vicinity of a point of of interest.

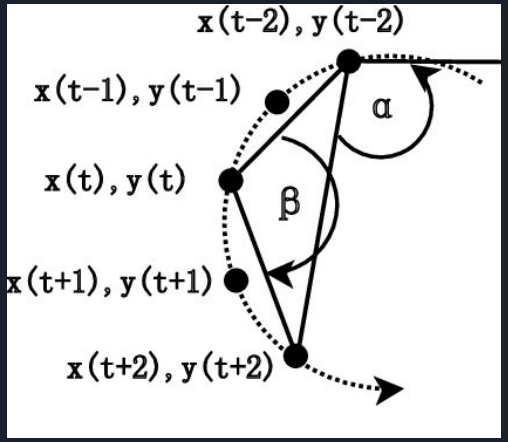
   Example: Vicinity slope of a point (x(t),y(t)) is the angle between the horizontal line i.e. x-axis and the line joining the points (x(t−2),y(t−2)) and (x(t+ 2),y(t+ 2)).

3. **Curvature β**
   Gives model an idea of amount of curvature that exists around a given point.

   Example: Curvature of a given point (x(t),y(t)) is the angle between the straight line joining (x(t−2),y(t−2)) and(x(t),y(t)) and straight line joining (x(t+ 2),y(t+ 2)) and (x(t),y(t)).

4. **Normalized y-coordinates**
   Transformed the y co-ordinate's range to be [0, 1] while preserving the width-height aspect ratio.

Fig. 1: Slope and curvature



[1] L. Hu and R. Zanibbi. 2011. HMM-Based Recognition of Online Hand-written Mathematical Symbols Using Segmental K-Means Initialization and a Modified Pen-Up/Down Feature. In 2011 International Conference on Document Analysis and Recognition. 457–462.https://doi.org/10.1109/ICDAR.2011.98

Parser

The parser searches the space of possible SLT outputs using the following steps:

1. Calculate distance matrix storing minimum distance between each pair of symbols
2. Build the minimum spanning tree
3. Get edges of the minimum spanning tree
4. Find out relationships between pair of symbols in each edge of the tree
    a. Find out if the symbols overlap vertically and horizontally after applying error margin
    b. If symbols overlap only horizontally, then label relationship as Right/Left based on direction of horizontal overlap
    c. If symbols overlap only vertically, then label relationship as Above/ Below based on direction of vertical overlap
    d. If symbols overlap both vertically and horizontally, then label relationship as Inside
    e. If symbols doesn't overlap either vertically or horizontally, then label relationship as Sup/Sub based on direction of vertical overlap
5. Write relationship in .lg file

Parser

Symbol Relationship Identification using overlap and positioning of traces

We implemented three variations of algorithms for relationship identification

1. Identification based on all the traces of a symbol
2. Identification based on minimum and maximum values of x and y coordinates of the traces
3. Identification based on average of traces in a particular symbol traces

Parser

Fig 1. Right Relationship

Fig 2. Inside Relationship

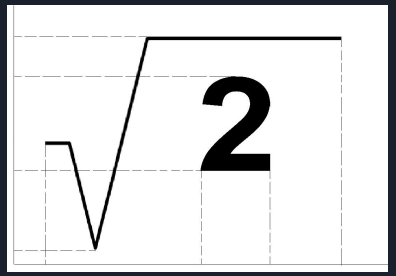


Fig 3. Sup Relationship

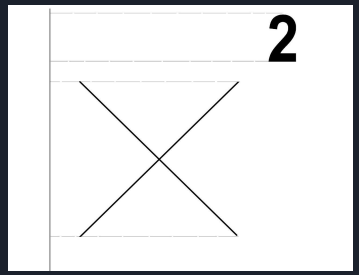
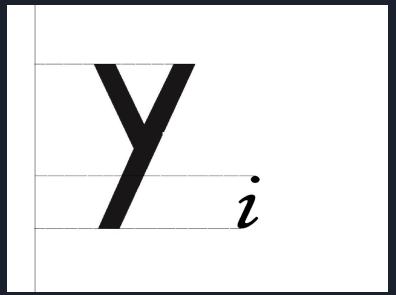


Fig 4. Sub Relationship



Results

		Objects						Files	
		Relationship			Relationship + Classes			Structure	Structure + Classes
		Recall	Precision	F-measure	Recall	Precision	F-measure	Rate	Rate
Left-Right Oracle	Stroke Oracle	29.79%	40.75%	34.42	29.79%	40.75%	34.42	9.25%	9.27%
	Symbol Oracle	72.64%	100.00%	84.15	72.64%	100.00%	84.15	40.73%	40.84%
Spanning Tree Oracle	Stroke Oracle	34.85%	40.60%	37.51	34.85%	40.60%	37.51	9.82%	9.86%
	Symbol Oracle	83.46%	100.00%	90.9	83.46%	100.00%	90.9	45.98%	46.32%
Our Parsing Model		38.46%	42.23%	40.25	38.46%	42.23%	40.25	11.23%	11.23%

Error Analysis and Limitations

- The errors obtained in final model is the result of errors generated stochastically from segmentation, classification and parsing.
- Our segmentation model was not the most optimal one and since our parsing models are based on geometric values, it will lead to misclassification of relationships among symbols
- Parser based on all traces
  - The model is prone to errors when the symbols are not ideally located on left, right, above or below positions.
  - If the overlap is more than a threshold, it will trigger false overlaps in the relationship identification algorithm
  - It is difficult to find a common overlap error margin threshold empirically
  - The parser was more prone to ‘Inside’ relationships
- Parser based on min max values
  - This parser is also prone to errors when the symbols are not ideally located on left, right, above or below positions.
  - If there is much overlap between the symbols, it will lead to false identification of overlap
  - Due to this, the parser classifies many relationships as ‘Inside’
- Parser based on average values of traces
  - This model is more prone to irregular shapes of symbols
  - Irregular shapes have outlier traces which might move the average coordinate to some other place than where they should be.

Thank You