UROPS 19/20 Report
Optical Character Recognition for Math Equations
Project Code 1920-00054

Gao WenHan A0172770A
Supervisor: Vikneswaran S/O Gopal Senior Lecturer

April 1, 2020

Abstract

This is a project which attempts to develop a pipeline for detecting handwritten mathematics equations, and then assessing their correctness through automatic grading. The foundation of this project the *Seshat* parser developed by Francisco Álvaro [1]. We also make use of properitory softwares and libraries such as *Xournal++* and *Sympy* in the construction of the pipeline architect. In this work, we mainly focus on mathematical expression equality and also solving for a variable.

Contents

1 Introduction and Motivation **1**

2 Literature Review **2**
2.1 SCG Ink . . . 2
2.2 Seshat Model . . . 2
2.2.1 Statistical Framework . . . 2
2.2.2 Symbol Likelihood . . . 2
2.2.3 Structural probability . . . 3
2.2.4 Parsing Algorithm . . . 3

3 Pipeline Architecture **4**
3.1 Data Input . . . 4
3.2 Data Processing . . . 4
3.3 Data Output . . . 4

4 Reviews **4**
4.1 Limitations . . . 4
4.2 Alternatives . . . 4

1 Introduction and Motivation

In most mathematics classes today, a teacher is responsible for marking many similar scripts. This manual process can be very repetitive. As artificial intelligence continous to development, we seek to explore a way that can automate this process. Mathematics questions can be broadly separated into two categories. The first is expression equality where one needs to show expression on the lefthand side is equal to the righthand side (1), using a series of equal expressions which links both together.

$$x^2 + 2x = (x+1)^2 - 1 \tag{1}$$
$$x^2 + 2x = x^2 + 2x + 1 - 1 \tag{2}$$
$$= (x+1)^2 - 1 \tag{3}$$

The other category is variable solving where one is given an equation with unknown variable(s) (4) and needs to solve for the unknown.

$$2x + 1 = 3 \tag{4}$$
$$x = 1 \tag{5}$$

In this project, we will be focusing on these two categories of mathematics questions and propose an end-to-end data processing pipeline that will perform automated grading for them.

2 Literature Review

2.1 SCG Ink

The SCG ink format is a text-based representation of the (x, y) coordinates of the strokes drawn by the user [2]. Each SCG ink file contains the total number of strokes appearing. For each stroke, there is a number indicating the number of points appearing in that stroke and all the (x, y) coordinates of those points. Therefore, it is able to capture the sequential relationships of the input strokes as compare to the conventional pixel image. The parser takes a SCG ink file as input and makes use of this sequential infomation of the strokes in its process to determine the mathematical symbols.

2.2 Seshat Model

2.2.1 Statistical Framework

The paper [1] propose modelling the structureal relationship of a mathematical expression using a statistical grammatical model. The authors define the problem as obtaining the most likely pasrse tree given a sequence of strokes of mathematics input. The tree derivation will produce the sequence of pre-terminals that represents the structure that relates all the mathematical symbols. In order to generate the sequence of pre-terminals, the parser will take into account all the strokes combinations to form possible symbols.

Therefore, the problem can be futher divided into two. First, the segmentation and recognition of symbols. Second, the structural analysis of the maths expression. We can see that these two problems are closely related and are interdependent. Hence, the proposed strategy computes the most likely parse tree while smultaneously solving for symbol segmentation, recognition and structural analysis of the input. More formally, the problem is defined as such

$$\hat{t} \approx \underset{t \in \mathcal{T}}{\operatorname{argmax}} \max_{\mathbf{s} \in \mathcal{S}; \mathbf{s} = yield(t)} p(\mathbf{s}|\mathbf{o}) \cdot p(t|\mathbf{s})$$

where $\mathbf{o}$ is the sequence of input strokes, $\mathbf{s}$ is the sequence of mathematical symbols derived from the strokes, and $\hat{t}$ is the most likely parse tree. This defnition assumes that the structural part of equation only depends on the sequence of the pre-terminal $\mathbf{s}$. $p(\mathbf{s}|\mathbf{o})$ represents the symbol likelihood and $p(t|\mathbf{s})$ represents the structural probability.

2.2.2 Symbol Likelihood

A symbol is made up of one or more strokes. In the paper, the authors define a symbol likelihood model that is not based on time of the information but rather the spatial information as symbols can be made up using non-consecutive strokes. However, it is reasonable to believe that only strokes that are close to

one another will form a mathematical symbol and this will greatly reduce the search space for the possible symbol segmentations. Overall, the authors futher factor the symbol likelihood into three models: a symbol segmentation model, a symbol classification model and a symbol duration model.

The symbol segmentation model uses the concepts of visibility and closeness of the strokes to generate a graph G with each stroke as a node and the edges only connect strokes that are visible and close. Then a segmentation hypothesis is valid only if the strokes it contains form a connected subgraph in G. This is able to reduce the set of possible strokes segmentation. Then for each stroke in a possible segmentation, geometric features are generated, such as mean horizontal position, mean veritical position etc. All these features are trained using a Gaussian Mixture Model (GMM).

The symbol classification model uses two Bidirectional Long Short-Term Memory Recurrent Neural Network (BLSTM-RNN) classifiers. RNNs remembers that past and it's decisions are influenced by what it has learnt from the past [3]. Unlike a normal NN where outputs are influenced by the weights applied to the inputs, there is also a hidden state vector in RNN that represents the context based on prior inputs or outputs. Hence, the same input can produce a different output due to diference in the previous inputs in the series. In handwriting recognition tasks where there could be considerable ambiguity given just one part of the input, a bidirectional RNN allows the network to access the context information in both time directions and detect the present. Lastly, Long Short-Term Memory is an advance architecture that allows cells to access context information over long periods of time.

Two classifier models are separated into online and offline. The online BLSTM-RNN uses online features of the points such as normalized (x, y) coordinates, normalized first and second derivatives etc. At the same time, the a binary image representation of the input is rendered and processed. Then several offline features of the image such as number of black pixels in the column, center of gravity of the column etc. The probabilities computed by these two classifiers are then combined using linear interpolation and a weight parameter.

Finally, the symbol duration model accounts for the intuitive idea that a mathematical symbol class is usually made up of a fixed number of strokes. Hence, a simple way to calculate the probability of of observing l_i strokes given a certain symbol class is proportion of the number of times that symbol class was composed using l_i strokes over the total number of that symbol class in the set.

2.2.3 Structural probability

The authors define a generative model based on a two-dimensional extension of the well-known context-free grammatical models to compute the parse tree probability. Context-free model is able to represent the structure of natural language and hence can be extented in to mathematics notations. This is because there are also structural dependencies between different elements in the mathematics expression. The parse tree probability can be factor into the probability of the rules of the gramma and the probability of two region that can be arranged according to a spatial relationship r. A spartial relationship model is used to model the later probability. Given two regions, the authors defined nine geometric features and then trained a GMM to compute the posterior probability for each spatial relationship r given any two regions. This posterior probability is then offset by a penalty function based on the minimum distance between the two regions.

2.2.4 Parsing Algorithm

The parsing algorithm is a dynamic programming method. At the initialization step, the probability of every valid segmentation is computed. At the general step, the parsing algorithm computes a new hypothesis by merging perviously computed hypotheses until all strokes are parsed. Then the most likely hypothesis and its associated derivation tree $\hat{t}$ is retrieved.

3 Pipeline Architecture

3.1 Data Input

3.2 Data Processing

3.3 Data Output

4 Reviews

4.1 Limitations

4.2 Alternatives

References

[1] Francisco Álvaro, Joan-Andreu Sánchez, and José-Miguel Benedí. "An integrated grammar-based approach for mathematical expression recognition". In: *Pattern Recognition* 51 (2016), pp. 135 –147. ISSN: 0031-3203.

[2] Scott MacLean et al. "Grammar-based techniques for creating ground-truthed sketch corpora". In: *International Journal on Document Analysis and Recognition (IJDAR)* 14.1 (2011), pp. 65–74.

[3] Mahendran Venkatachalam. *Recurrent Neural Networks*. 2019. URL: `https://towardsdatascience.com/recurrent-neural-networks-d4642c9bc7ce`.