2011 International Conference on Document Analysis and Recognition

CROHME2011: Competition on Recognition of Online Handwritten Mathematical Expressions

Harold Mouchère and Christian Viard-Gaudin
IRCCyN/IVC – UMR CNRS 6597
Ecole Polytechnique de l'Université de Nantes
44306 Nantes Cedex 03, France
{Harold.Mouchere,Christian.Viard-Gaudin}@univ-nantes.fr

Dae Hwan Kim and Jin Hyung Kim
Division of Computer Science
Korea Advanced Institute of Science and Technology
Seongbuk-gu, Seoul 136-791, Republic of Korea
ofreunde.edward@gmail.com, jkim@kaist.edu

Utpal Garain
Computer Vision and Pattern Recognition (CVPR) Unit
Indian Statistical Institute,
Kolkata 700108, India
utpal@isical.ac.in

Abstract— **A competition on recognition of online handwritten mathematical expressions is organized. Recognition of mathematical expressions has been an attractive problem for the pattern recognition community because of the presence of enormous uncertainties and ambiguities as encountered during parsing of the two-dimensional structure of expressions. The goal of this competition is to bring out a state of the art for the related research. Three labs come together to organize the event and six other research groups participated the competition. The competition defines a standard format for presenting information, provides a training set of 921 expressions and supplies the underlying grammar for understanding the content of the training data. Participants were invited to submit their recognizers which were tested with a new set of 348 expressions. Systems are evaluated based on four different aspects of the recognition problem. However, the final rating of the systems is done based on their correct expression recognition accuracies. The best expression level recognition accuracy (on the test data) shown by the competing systems is 19.83% whereas a baseline system developed by one of the organizing groups reports an accuracy 22.41% on the same dataset.**

Keywords-Evaluation; Mathematical expressions; Online handwriting, symbol recognition.

I. INTRODUCTION

Pioneering attempt towards automatic recognition of handwritten mathematical expressions dates back to 60's of the previous century [1]. After this initial attempt, several researchers have tried to study this problem at different paces [2]. However, during the last decade this research has gained considerable attention from the research community. There are several reasons behind this renewed interest. As an application, online recognition of expression provides a better human computer interface in order to prepare scientific documents. If successful system can be developed, entry of mathematics in documents would be easy. On the other hand, as research problem recognition of handwritten mathematics exhibits several fascinating challenges. The recognition problem is different from the traditional OCR problem. Correct parsing of two-dimensional structure of an expression is not only an interest of the OCR community but also of many researchers from other fields. Presence of enormous uncertainties and ambiguities makes the understanding problem difficult and at the same time enticing for the researchers. Achieving success in this domain would progress the state of the art in understanding of visual languages.

Therefore, many researchers around the world have been studying this problem, i.e. recognition of mathematical expressions. Every year, several papers are published in related journals, many papers are presented in relevant conferences. On this particular problem, altogether, more than 150 contributory papers have already been published in different journals and conference proceedings. In spite of this effort, it is very difficult to bring out a state of the art progress of this research. This is due to the fact that most of the research groups have been presenting their results based on their own dataset. Oftentimes, these datasets are not sharable and not available in the public domain. Therefore, one group cannot replicate results of others and hence, cannot clearly judge their progress too. Overall, the advancement of the field remains grey.

The proposed competition named as CROHME (Competition on Recognition of Online Handwritten Mathematical Expressions) is aimed at bringing the researchers under a common platform so that they share the same dataset for their respective research and report performance of their systems on a common test data. The outcome of this event not only documents the advancement and challenges of the relevant research but at the same time, the individual group understands relative strength and shortcomings of their system with respect to those of others. This will also serve as a ready reference to other researchers and practitioners working in this area or the new comers. As most of the researchers working in this area belong to the community participating in the Int. Conf. on Document Analysis and Recognition (ICDAR), we choose ICDAR 2011 as the right platform to hold the competition.

1520-5363/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDAR.2011.297

The rest of the paper is organized as follows. Section-2 provides an overview of the competition, its organizers, the participants, data set, evaluation strategies, etc. Section-3 gives elaborate information on data format and organization of the data set and its content and coverage. Section-4 briefly describes the working methodologies of the participating systems. Section-5 presents the evaluation results and announces the winner of this competition. The following section, i.e., section-6 concludes the paper.

II. OVERVIEW OF THE COMPETITION

The competition is organized by the three research labs (one from France and the other two from Asia) to which the authors of this paper are affiliated. Six research groups registered themselves for participating in this event. Finally, four research groups submitted their systems. The competition was held among these four systems. The fifth system was developed by one of the organizing groups and hence, it did not join the race but the performance of this system is worth mentioning as this serves as a baseline system.

As the data for online handwritten expressions consists of several information, a markup language (i.e. InkML format) is first defined to clearly explain how expression data is stored. Two parts are defined in the training dataset. Part-I contains 296 expressions whereas Part-II is consisting of 921 expressions. The part-II set includes the part-I expressions. The reason behind dividing the expression set into two is to grade the expressions as per their complexity in terms of number of distinct symbols and the types of mathematical operations used in them. Part-I expressions are less complex than those of Part-II in sense that number of distinct symbols used in Part-I expressions is less than that of Part-II. Also, less variation is allowed in using different mathematical operations for Part-I expressions. Each part is characterized by its underlying grammar. The respective grammar defines the types of expressions one may expect in Part-I or Part-II. The details about the grammars are given in the next section.

Test dataset is different from the training set. Test expressions are also divided into two parts conforming to the grammars defined for each one. Part-I of the test set contains 181 expressions whereas Part-II consists of 348 expressions including the part-I samples. The training data was distributed two and half months before the evaluation of the systems. Instead of distributing the test dataset, the participating groups were advised to submit their systems to the organizers. Testing was done at the organizers' end. Four parameters as explained in section-5 are measured for evaluating the recognizers. However, the final rating is done based on the expression recognition accuracy. The details of evaluation are reported in section-5.

III. DATA FORMAT, TRAINING AND TESTING DATA SETS

The ink corresponding to each expression is stored in an InkML file. An InkML file mainly contains three kinds of information: (i) the ink: a set of traces made of points; (ii) the symbol level ground truth: the segmentation and label information of each symbol of the expression; and (iii) the expression level ground truth: the MathML structure of the expression.

The two levels of ground truth information (at the symbol as well as at the expression level) are entered manually. Furthermore, some general information is added in the file: (i) the channels (here, X and Y); (ii) the writer information (identification, handedness (left/right), age, gender, etc.), if available; (iii) the LaTeX ground truth (without any reference to the ink and hence, easy to render); (iv) the unique identification code of the ink (UI), etc.

The InkML format enables to make references between the digital ink of the expression, its segmentation into symbols and its MathML representation. An example of an InkML file for the expression $a < b / c$ is shown below. It contains 6 strokes for 5 symbols (two for the 'a', and one for each of the other symbols). Note that the traceGroup with identifier xml:id="8" has references to the 2 corresponding strokes of symbol 'a', as well as to the MathML part with identifier xml:id="A". Thus, the stroke segmentation of a symbol can be linked to its MathML representation.

AN EXAMPLE InkML FORMAT

```
<ink xmlns="http://www.w3.org/2003/InkML">
<traceFormat>
<channel name="X" type="decimal"/>
<channel name="Y" type="decimal"/>
</traceFormat>
<annotation type="writer">w123</annotation>
<annotation
type="truth">$a<\frac{b}{c}$</annotation>
<annotation  type="UI">  2011_IVC_DEPT_F01_E01
</annotation>
<annotationXML    type="truth"    encoding    =
"Content-MathML">
<math
xmlns="http://www.w3.org/1998/Math/MathML">
<mrow>
<mi xml:id="A">a</mi>
<mrow>
<mo xml:id="B"><</mo>
<mfrac xml:id="C">
<mi xml:id="D">b</mi>
<mi xml:id="E">c</mi>
</mfrac>
</mrow>
</mrow>
</math>
</annotationXML>
<trace id="1">985 3317, ..., 1019 3340</trace>
       ...
<trace id="6">1123 3308, ..., 1127 3365</trace>
<traceGroup xml:id="7">
<annotation                type="truth">Ground
truth</annotation>
<traceGroup xml:id="8">
<annotation type="truth">a</annotation>
<annotationXML href="A"/>
<traceView traceDataRef="1"/>
<traceView traceDataRef="2"/>
</traceGroup>
       ...
</traceGroup>
</ink>
```

As mentioned before; both the training and test datasets are divided into two parts. The type of expressions that are allowed to appear in a part is dictated by the corresponding grammar, which is applied on the LaTex string. The grammar for Part-I samples of the data accepts only 37 terminals: (i) 10 digits (0-9), (ii) 11 letters (a, b, c, d, e, i, k, n, x, y, z), (iii) 3 Greek letters (phi, theta, pi), (iv) 2 function words (sin, cos), (v) 2 structure symbols (root, i.e. *sqrt* and fraction, i.e. *frac*), (vi) 3 operator symbols (+, -, *pm*), (vii) 4 relational operator symbols (=, *neq*, *leq*, *lt*), and (vii) 2 parenthesis symbols (‘(‘ and ‘)’).

For Part-I, the limitations among logical relationships are: (i) only one symbol in subscript or superscript is allowed, (ii) no recursive fraction is there, however, a sum of fraction or fraction of sum may appear but no fraction of fractions can be allowed, (iii) no product of fraction can appear in the expressions. Two permissible recursive expressions are: (a) repeated sum, i.e. sum of sums is permitted and (b) nested root, i.e. a square root can be found in other square root.

The grammar for Part-II expression is less restricted than that of Part-I. The number of terminal symbols is increased to 57. The new (in addition to Grammar-I symbols) terminals are: (i) 5 letters (A, B, C, F, j), (ii) 3 Greek letters (alpha, beta, gamma), (iii) 2 function words (tan, log), (iv) 2 operators (div, times), (v) 1 relational operator (geq), (vi) 2 elastic operators (sum, int), (vii) 3 structural operators (two for limit structures, *lim* and *rightarrow*; one for factorial, i.e. !), (viii) 2 special symbols (infinity, i.e., *infty* and dots, i.e. *cdots* and *ldots* are treated equally).

The number of restrictions on logical relationships is also less in Grammar-II. There are no limits on recursions of operations like sum, product, function call, fraction, root, sub/superscript on symbols, etc. However, a few restrictions are still there. For example, in case of using sub/superscript of function names, only one symbol is allowed. For explicit grammar rules one may look at the material available in the competition site, i.e. http://www.isical.ac.in/~crohme2011. Validators are available to check whether a given expression conforms to a particular grammar. The validator extracts the LaTeX string of the expression and parses it to validate whether it is accepted by a grammar (i.e. Grammar-I or Grammar-II).

The number of samples available with the training and test datasets are given in Table I. Note that Part-I is a subset of Part-II samples as Grammar-II always accepts the expressions generated by Grammar-I.

TABLE I: VOLUME OF TRAINING AND TEST DATA

Dataset Type	**Number of expression samples in**	
	Part-I	**Part-II**
Training	296	921
Test	181	348

IV. OVERVIEW OF THE PARTICIPATING SYSTEMS

Four participating groups submitted their systems but in total, five systems were evaluated. The fifth one developed by one of the organizing groups and hence, did not compete.

System-I: This is developed by Lei Hu, Richard Pospesel, Kevin Hart, and Richard Zanibbi of the Rochester Institute of Technology (NY, USA). A preliminary version of the system was evaluated as the group could not complete the revised version of their system in time to meet the extended deadline. The recognition architecture was feed-feedforward. The time-ordered sequence of strokes for an expression is broken down into groups of three; for *n* strokes, the system produces a list of stroke subsets, i.e. $S = ((s_1, s_2, s_3), \ldots, (s_{n-2}, s_{n-1}, s_n))$; if *n* is not divisible by 3, the final stroke subset may have only one or two strokes. For each three-stroke group, an HMM classifier using four features [3] is used to compute the highest classification probability for each possible segment, and then the segmentation that maximizes the harmonic mean of the resulting class probabilities is selected for the stroke group. Once symbols are segmented and recognized, the DRACULAE parser [4] recovers the expression structure using default parameters.

System-II: This system is developed at the Sabanci University, Turkey. The system uses a 2D-stochastic grammar. Each grammar rule decides whether the required relationships (up, down, inside, etc.) exist between two symbols. For each production rule, a likelihood score is assigned to the generated token based on the likelihood of the component symbols and the likelihood of the 2D relationship between the component symbols. Then, the best tokens are expanded until no more grammar rules can be applied. The system thus generates all likely interpretations of a given input string, along with their likelihoods. The details of this system can be found in [5].

System-III: This system is developed by Francisco Álvaro, Joan-Andreu Sánchez and José-Miguel Benedí of the Instituto Tecnológico de Informática. Universitat Politècnica de València. A stochastic parsing based on Probabilistic Context Free Grammar is used. First, Hidden Markov Models are used to obtain segmentation and symbol recognition hypotheses, by using both online and offline information. Then a CYK-based algorithm obtains the most probable parse according to a two-dimensional PCFG. Thus, this system is able to solve jointly all the steps involved in the mathematical expression recognition problem.

System-IV: This system is developed at the Institute for Language and Speech Processing, Athena Research Center, Greece and named as Math-ILSP system. The system incorporates the following four major modules: (i) symbol detection based on the spatial relations of the neighboring strokes, (ii) symbol recognition by applying the elastic matching algorithm described in [6], (iii) detection of the levels of an expression by examining the dominance relationships of the recognized symbols based on their spatial relations and special constrains for each symbol, and (iv) construction of the expression by employing the relations of the detected levels.

System-V: The fifth system is developed at IRCCyN- IVC, Université de Nantes, France. The system [7, 8] aims at handling mathematical expression recognition as a simultaneous optimization of symbol segmentation, symbol recognition, and 2D structure recognition under the restriction of a mathematical expression grammar. To achieve this, the system considers a hypothesis generation mechanism supporting a 2D grouping of elementary strokes, a cost function defining the global likelihood of a solution, and a dynamic programming scheme giving at the end the best global solution according to a 2D grammar and a classifier. As a classifier, a Neural Network architecture is used; it is trained within the overall architecture allowing rejecting incorrect segmented patterns.

V. EVALUATION

For each system, four aspects are evaluated. They are: (i) ST_{rec}: stroke-level classification rate, (ii) SYM_{seg}: symbol segmentation rate, (iii) SYM_{rec}: symbol recognition rate (for correct segments) and (iv) EXP_{rec}: expression-level recognition rate. All the above four rates are represented in percent (%). When the systems were evaluated by using Part-I samples, the results are reported in Table II and Table III reports the evaluation results when Part-II samples of the test dataset are used for evaluation.

TABLE II: EVALUATION WITH PART-I TEST DATA.

Systems	**ST_{rec}**	**SYM_{seg}**	**SYM_{rec}**	**EXP_{rec}**
System-I	53.23	59.06	88.78	4.42
System-II	22.39	27.98	82.11	0.55
System-III	**78.73**	**88.07**	**92.22**	**29.28**
System-IV	37.41	55.15	81.71	0.00
System-V	78.57	87.56	91.67	40.88

TABLE III: EVALUATION WITH PART-II TEST DATA.

Systems	**ST_{rec}**	**SYM_{seg}**	**SYM_{rec}**	**EXP_{rec}**
System-I	51.58	56.50	91.29	2.59
System-II	22.11	28.25	83.76	0.29
System-III	**78.38**	**87.82**	**92.56**	**19.83**
System-IV	52.28	78.77	78.67	0.00
System-V	70.79	84.23	87.16	22.41

It is noted that systems behaved very consistently for both the test datasets. As Part-II expressions are more complex than those in Part-I, systems' performances (i.e. EXP_{rec}) too degrade. As it was decided that the final rating of the systems will be done on their expression-level recognition accuracies, System-III turns out to be the winner of this competition. Note that System-V did not compete with others as it was developed by one of the organizing groups. This system can be considered as a baseline system.

VI. CONCLUSIONS

The need for organizing a competition on recognition of mathematical expressions was being felt for some time. Many research groups have been working in this area but unavailability of a common dataset and evaluation strategy made it difficult to understand the progress of this research. The present effort bridges this gap. The evaluation results show that though the systems are doing well in classifying strokes, segmenting symbols or even in recognizing symbols, expression level recognition (EXP_{rec}) accuracies are very poor. The presence of enormous ambiguities in understanding the 2D arrangement of symbols is the major reason behind such a low EXP_{rec}. This clearly shows a direction of the future research in this area. Some more versions of this competition must be conducted in future to attract more research groups and new recognition methods to improve the recognition accuracy.

Because of the resources generated under this competition, the present researchers as well as the new comers would be able to conduct the respective research in a systematic manner. The evaluation results reported here can be considered as a state of the art of this research. The datasets will remain available to the research community. Hence, many other groups who could not participate in this competition (or the new comers) can evaluate their systems using this dataset to check their systems' performances against the results reported in this paper.

REFERENCES

[1] R.H. Anderson, "Syntax-directed Recognition of Hand-printed Two-dimensional Mathematics," *Doctoral Dissertation*, Dept. of Engineering and Applied Physics, Harvard University, 1968.

[2] Utpal Garain, "Machine Recognition of Mathematical Expressions," Publisher: *Lambert Academic Publishing*, Germany, ISBN: 978-3-8433-6371-6, December, 2010.

[3] H. Lei and R. Zanibbi, "HMM-Based Recognition of Online Handwritten Mathematical Symbols Using Segmental K-means Initialization and a Modified Pen-up/down Feature," in Proc. ICDAR 2011 (to appear).

[4] R. Zanibbi, D. Blostein, and J.R. Cordy, "Recognizing mathematical expressions using tree transformation," IEEE Trans. Pattern Analysis and Machine Intelligence, 24 (11), 2002, 1455-1467.

[5] M. Celik and B. Yanikoglu, "Mathematical Formula Recognition Using a 2D Stochastic Graph Grammar," in Proc. ICDAR 2011 (to appear).

[6] F. Simistira, V. Katsouros, and G. Carayannis, "A Template Matching Distance for Recognition of On-Line Mathematical Symbols," in Proc. of the 11th International Conference on Frontiers in Handwriting Recognition (ICFHR), Quebec, Canada, 2008.

[7] Ahmad-Montaser Awal, Harold Mouchère, and Christian Viard-Gaudin, "Towards handwritten mathematical expression recognition," in Proc. of the 10th International Conference on Document Analysis and Recognition (ICDAR), Barcelone, 2009, 1046-1050.

[8] Ahmad-Montaser Awal, Harold Mouchère, and Christian Viard-Gaudin, "Improving online handwritten mathematical expressions recognition with contextual modeling," in Proc. of 12th International Conference on Frontiers in Handwriting Recognition (ICFHR), Calcutta, 2010.