Pattern Recognition Letters 172 (2023) 29–36

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

The IBEM dataset: A large printed scientific image dataset for indexing and searching mathematical expressions

Dan Anitei*, Joan Andreu Sánchez, José Miguel Benedí, Ernesto Noya

Pattern Recognition and Human Language Technologies Research Center, Universitat Politécnica Valéncia, Camino de Vera s/n, Valencia 46022, Spain

ARTICLE INFO

Article history:
Received 25 July 2022
Revised 26 May 2023
Accepted 30 May 2023
Available online 2 June 2023

Edited by: Maria De Marsico

Keywords:
Mathematical expression dataset
Mathematical expression recognition
Mathematical expression retrieval
Mathematical symbols classification

ABSTRACT

Searching for information in printed scientific documents is a challenging problem that has recently received special attention from the Pattern Recognition research community. Mathematical expressions are complex elements that appear in scientific documents, and developing techniques for locating and recognizing them requires the preparation of datasets that can be used as benchmarks. Most current techniques for dealing with mathematical expressions are based on Machine Learning techniques which require a large amount of annotated data. These datasets must be prepared with ground-truth information for automatic training and testing. However, preparing large datasets with ground-truth is a very expensive and time-consuming task. This paper introduces the IBEM dataset, consisting of scientific documents that have been prepared for mathematical expression recognition and searching. This dataset consists of 600 documents, more than 8 200 page images with more than 160 000 mathematical expressions. It has been automatically generated from the LaTeX version of the documents and can be enlarged easily. The ground-truth includes the position at the page level and the LaTeX transcript for mathematical expressions both embedded in the text and displayed. This paper also reports a baseline classification experiment with mathematical symbols and a baseline experiment of Mathematical Expression Recognition performed on the IBEM dataset. These experiments aim to provide some benchmarks for comparison purposes so that future users of the IBEM dataset can have a baseline framework.

1. Introduction

One of the most usual activities researchers, scientists, and scholars engage in worldwide is searching through the ever-increasing volume of science, technology, engineering, and mathematics (STEM) documents that online digital libraries constantly publish. The search for Mathematical Expressions (MEs) in a printed text has received recent attention, showing interest in integrating mathematical notation with textual information for the problem of Information Retrieval, as some competitions have demonstrated [1].[1] Plain text searching in large digital libraries of STEM documents does not constitute a challenge anymore, however searching for chemical formulas, maps, mathematical expressions, or other complex structures remains scarcely explored [2,3]. Searching for these complex structures in STEM documents cannot be approached like searching for plain text given the 2-dimensional structural information they convey. This 2-dimensional structural information involves semantic information that can not be simply solved with string-matching algorithms, requiring powerful Pattern Recognition and Machine Learning techniques that can provide a correct interpretation. This paper researches the construction of a large dataset for advancing in the problem of indexing and searching for ME in massive collections of digital STEM documents [4].

Two important problems arise when searching for MEs in digital documents. First, locating MEs and classifying them: embedded expressions are referred to as *inline* MEs, while isolated ones are referred to as *displayed* MEs. Since displayed MEs are separated from surrounding text, they can be easily located with profile projection methods. However, these expressions can be confused with other graphical elements found in STEM documents, such as plots, tables, or figures. Inline MEs are significantly more difficult to identify since they are easily confused with running text. The second problem is related to the interpretation process of the MEs through a parsing process.

* Corresponding author.
E-mail addresses: danitei@prhlt.upv.es (D. Anitei), jandreu@prhlt.upv.es (J.A. Sánchez), jmbenedi@prhlt.upv.es (J.M. Benedí), noya.ernesto@prhlt.upv.es (E. Noya).

[1] https://www.ntcir-math.nii.ac.jp/introduction/.

https://doi.org/10.1016/j.patrec.2023.05.033

Machine Learning methods are the state-of-the-art technology for detection [3,5,6] and recognition [5,7,8] of MEs in STEM document images. However, these methods require large amounts of annotated data. Considering that the ground-truth (GT) data is manually prepared and/or validated, the creation of annotated datasets incurs high costs and is a commonly recognized bottleneck in Machine Learning research. Preparing large freely available datasets with GT is a real need. Thus, the main motivation of this paper is to introduce a GT-rich dataset for ME recognition and, ultimately, for ME indexing and searching. The dataset contains GT for ME detection, mathematical symbol classification, and ME recognition.

To facilitate progress in ME recognition, in this paper, we also present baseline experiment results, obtained with open-source freely-available ME recognition systems [7,9,10]. These results are analyzed separately for inline and displayed MEs, emphasizing the different characteristics of each expression type. In addition, results of ME recognition with automatic ME detection methods are also presented.

The preparation of a dataset like the one presented in this paper has to overcome several difficulties and/or requirements: 1) copyright issues since many scientific documents are under some type of license; 2) the dataset has to be large since the variability in the ME is expected to be significant as it depends on the topic, field and area, on the author, on the type of documents (e.g. articles or slides), on the publisher requirements, etc.; 3) the scalability of the dataset in time, with the possibility of enriching the GT with as much information as possible would be desired; and 4) the dataset should be able to be processed automatically although human inspection of the final GT is mandatory. Due to these problems, we consider the first one as the most stringent, since all the others cannot be tackled without solving the first.

Having said that, this paper introduces the IBEM dataset[2] of printed scientific documents for ME searching and recognition research, currently consisting of 600 STEM papers, with more than 8 000 page images, containing more than 160 000 ME. The dataset was prepared automatically from a public set of documents. The preparation of the GT for each document was carried out from the LaTeX version. This makes this dataset scalable to include thousands of documents that can be used for researching and developing efficient recognition and searching techniques. Our goal was to create a large enough dataset to be used for training algorithms for recognition and searching ME in documents for which only the digital image is available.

This paper is organized as follows: In Section 2, existing ME datasets are reviewed. Then, Section 3 describes the design of the IBEM corpus, detailing the acquisition and post-processing of the data. In Section 4, an analysis of the results of the symbol classification and ME recognition experiments are presented. Finally, Section 5 summarizes the work presented.

2. Related work

The IM2LATEX-100K dataset [7], is a ME collection comprising of 103 556 different LaTeX MEs, for which the rendered images are provided. The corpus was built based on regular expressions, in which mathematical formulas were extracted by parsing the LaTeX sources of documents from the 2003 KDD cup [11]. The MEs extracted were then filtered by removing expressions that did not compile due to unmet dependencies or custom macros. MEs with less than 40 symbols were also removed. Although useful for investigating ME recognition, the IM2LATEX-100K GT does not include information enabling ME detection or searching experiments.

Table 1
Existing datasets statistics vs. IBEM dataset. Some figures are not applicable (n/a) or not specified (n/s).

Dataset	#Doc.	#Pages	#ME	#Symb.
Im2Latex	n/a	n/a	103 556	n/s
UW-III	n/s	1 600	≈ 100	n/s
InftyCDB	30	467	21 056	157 058
InftyMCCDB	30	467	19 381	142 063
GTDB-I	31	544	n/s	162 406
GTDB-II	16	343	n/s	115 433
IBEM	600	8 272	166 692	1 109 926

The UW-III collection [12], with GT that contains ME information, comprises of 1 600 scanned images of English documents with manually edited GT, for which the LaTeX version is also available. For each image, various bounding boxes have been included and marked based on the content (e.g. text, math, table, etc.). While the noise introduced into the dataset by including scanned copies of the original documents can be addressed [13], the limited size of the collection would make it less suited for research on ME detection and recognition.

The InftyCDB dataset [14], contains 21 056 MEs. Multiple versions of this dataset exist (InftyCDB-{1-3} and InftyMCCDB-{1-2}), in which GT is provided at symbol level, with the relationship among mathematical symbols having been defined manually. The dataset has been greatly explored for offline math symbol recognition experiments [15–17]. However, the GT was sourced from articles that weren't copyright-free, and therefore are not provided with the dataset.

The copyright limitation, which also affects the UW-III dataset, hinders the use of these collections for researching ME detection, as pointed out in [3]. The authors proposed a new GTDB dataset[3], which comprised of two versions, contains 47 articles of 887 pages. These articles include diverse font faces and mathematical notation styles, without providing the total number of ME. Finally, it is important to note that this dataset was utilized in a ME detection competition [4].

As a result, it is necessary to create a dataset that includes a large number of images of scientific article pages with localized and annotated MEs, as well as the associated mark-up language. Characteristics of the existing datasets in comparison to the IBEM dataset are shown in Table 1.

It is important to remark that the GT regarding the ME location of the IBEM dataset[4] presented in this paper has been successfully employed in the *ICDAR 2021 Competition on Mathematical Formula Detection* [18]. While excellent results were obtained in the detection of displayed ME, the results of this competition showed that there is still a margin for improvement, especially for inline MEs [19].

3. The IBEM dataset

3.1. Design of IBEM dataset

The IBEM corpus has been created considering the following design criteria: 1) it should be a massive collection of digital STEM document images, 2) it should be publicly available, and 3) it should be available in a format that allows automatic processing for preparing the GT.

[2] http://ibem.prhlt.upv.es/en/.

[3] https://www.github.com/uchidalab/GTDB-Dataset.

[4] https://www.zenodo.org/record/4757865.

Table 2
Statistics about the dataset.

Total no. of documents	600
Total no. of pages	8 272
No. of displayed MEs	29 603
No. of inline MEs	137 089

As a result, we chose the KDD Cup collection [11], which is publicly available, with the LaTeX sources of all documents provided. The KDD Cup collection has a large number of research papers (approximately 29 000) ranging from 1992 to 2003. Of these papers we selected 2 791, filtering out documents that did not compile and keeping documents from 2000 onwards to avoid compatibility issues with older versions of LaTeX libraries potentially used by authors.

LaTeX allows defining macros to abbreviate the formal notation and simplify typesetting. However, having LaTeX standard delimiters renamed increased the complexity of detecting MEs in the LaTeX source. Out of a total of 2 791 documents, 957 documents did not rename these delimiters, of which we manually selected 600 documents that required minimal correction. Table 2 summarizes the characteristics of the resulting dataset.

The GT of the IBEM dataset consists of:

- The LaTeX transcript of inline and displayed MEs.
- Coordinates of the bounding boxes enclosing the definition of ME in the rendered images.
- Images of individual pages contained in the documents that were processed.

An example of the previous output described can be seen in Fig. 1. Note that standalone images of MEs can be easily obtained by rendering the LaTeX transcripts.

3.2. Ground-Truth preparation

This section presents the process of extracting the GT from LaTeX documents, with the challenges that arose during implementation and the respective adopted solutions.

To obtain the GT and the result shown in Fig. 1, the extraction process was divided into two parts. The first part focused on the design of LaTeX macros to highlight the MEs. The second part consisted of automating the extraction and generation of the ME LaTeX GT. In this second phase, regular expressions were designed to detect the delimiters used to define LaTeX mathematical environments. Through these regular expressions, the macros defined in the first phase were embedded into the mark-up language of MEs. For this reason, we decided to focus only on documents that did not rename these delimiters.

Most of the challenges we faced were due to the flexibility of LaTeX typesetting and the high variability in the definition of MEs expected when working with extensive collections of documents, making the use of regular expressions more difficult. More specifically, we had to tackle the following problems:

1. There are two types of MEs, inline and displayed. The formulae are rendered accordingly to the surrounding elements, increasing the difficulty of detecting their location.
2. ME mark-up language can run on more than one line, making the correct detection of the definition of these expressions more complex.
3. Parts of the GT, could not be extracted from within LaTeX, due to the page ship-out routine of LaTeX that could cause changes after the extraction phase.
4. The documents in the dataset included a large number of delimiters for LaTeX mathematical environments that would most likely increase when processing new documents, considering that LaTeX is constantly evolving.

3.2.1. ME Bounding box detection

As mentioned, the first challenge we faced was related to the geometric position of MEs. This required dealing with certain special cases, such as: 1) a ME can run on more than one single line, on different pages, and/or on different columns; 2) an inline ME can appear in a caption, in a footnote, in a drawing, or in a plot; 3) a displayed ME can include a numbering. All these situations were considered in the processing of the dataset.

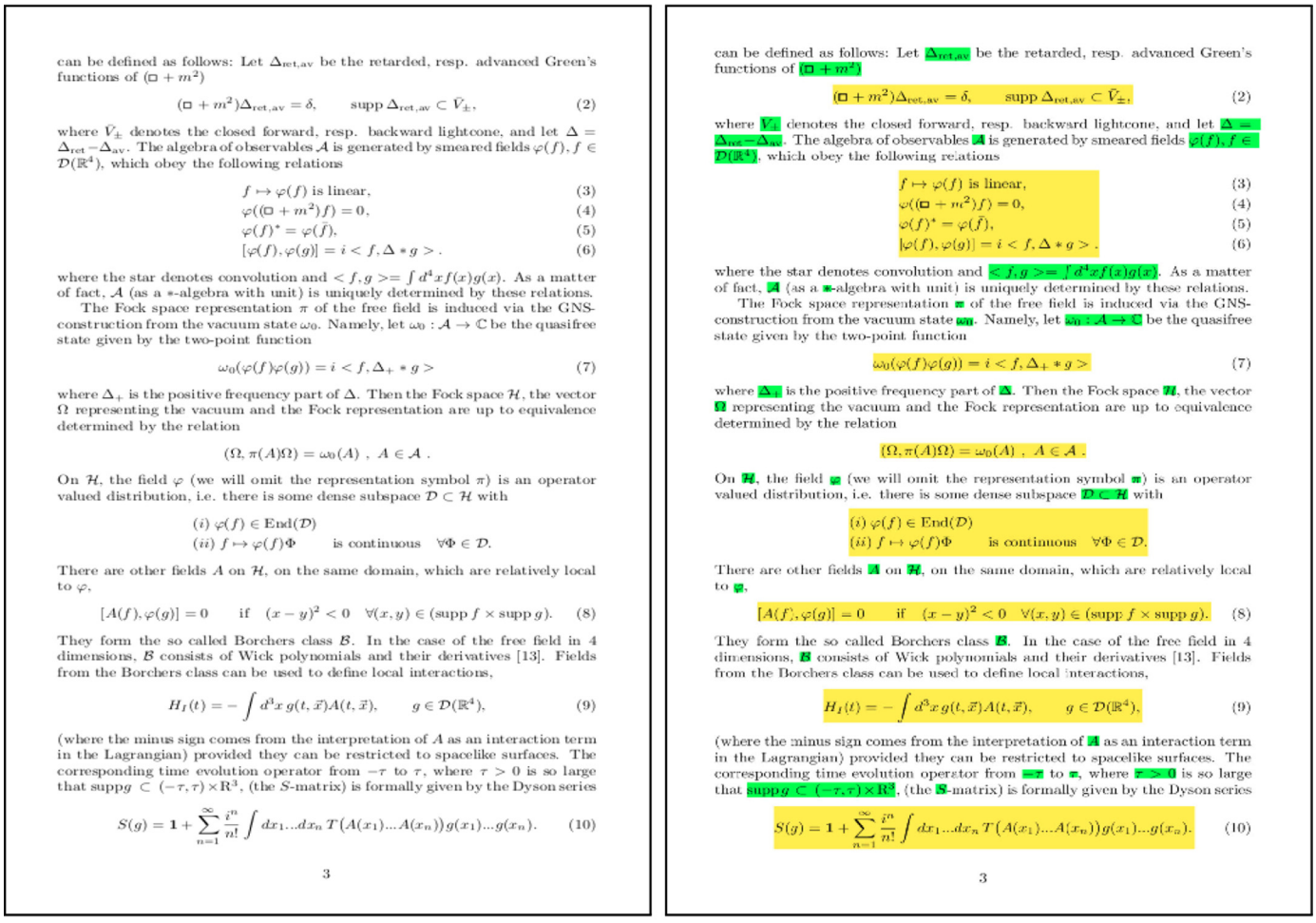
To determine the location of the bounding boxes of MEs, coordinate measurements[5] were taken at the first and last mathematical symbol of each formula. MEs were then highlighted [6] for future extraction.

In the case of inline MEs, two coordinates proved sufficient for determining the shape of the bounding boxes. By measuring the distance in the x and y axis of these two points, we could establish whether these expressions have been rendered on one line or more, or even if they were split over two pages, each case being treated accordingly. For displayed MEs, the first and last symbols of the expression were not always rendered in the upper left and lower right corners of the bounding box. Fig. 2 provides an example of such a situation. To account for such situations, macros were embedded to take coordinate measurements before and after newline symbols and between symbols of superscript and subscript elements, which, when rendered, are vertically shifted relative to the baseline. As in the case of inline ME, the problem of displayed ME split over two pages or columns was addressed by detecting large gaps between consecutive coordinate points pertaining to the same ME.

3.2.2. Automating the construction of the ME LaTeX GT

To automate the construction of the GT, we defined regular expressions[7] to extract the mark-up language of MEs and to embed the LaTeX highlighting macros into the LaTeX source files of the documents. To address the problem of mark-up language detecting for MEs typeset over multiple lines, pattern matching was done at the paragraph level. The LaTeX source files were compiled and split into images of the corresponding pages. Each page was passed through a yellow and green color filter[8] to extract the coordinates of bounding boxes of MEs. At this point, ground-truth correction was done by manually inserting highlighting macros where regular expressions failed to detect a mathematical environment. Also, bounding boxes were manually adjusted where partial clipping of symbols was detected.

[5] Zref package: https://www.ctan.org/pkg/zref?lang=en.
[6] Tikz package: https://www.tikz.dev/.
[7] SED: https://www.gnu.org/software/sed/manual/sed.html.
[8] OpenCV: https://www.opencv.org/.



(a) Original page. (b) Highlighted bounding boxes.

Fig. 1. An example of processing a page of the IBEM dataset. Inline ME are highlighted in green, while displayed ME are highlighted in yellow.

$$\sum_{Y_1\sqcup Y_2=Y,\ X_1\sqcup X_2=X} (-1)^{(|Y_1\cap Y_4|+|X_1\cap X_4|)}[R^*(Y_1\cap Y_4, X_1\cap X_4)\times_\hbar A\times_\hbar R(Y_2\cap Y_4, X_2\cap X_4)]\cdot (-1)^{(|Y_1\cap Y_3|+|X_1\cap X_3|)}[R^*(Y_1\cap Y_3, X_1\cap X_3)\times_\hbar R(Y_2\cap Y_3, X_2\cap X_3)] = 0. \quad \square$$

Fig. 2. Example of a displayed ME for which detecting the first and last symbol is not sufficient for correctly computing the bounding box of the ME. The incorrectly computed bounding box is highlighted in yellow.

3.3. LaTeX *transcript normalization*

Considering that the purpose of creating the IBEM dataset is to aid in ME recognition and retrieval research, having user-defined macros in the GT would hinder the accuracy of these systems. To address this issue, each ME was converted into an abstract syntax tree (AST), based on the LuaLaTeX `nodetree`[9] package. At this level, the LaTeX mark-up language has been transformed into node lists, containing the glyphs to be printed. Through a depth-first traversal of the AST, the MEs were reconstructed by mapping the Unicode code points of the glyphs to the LaTeX commands that could render them. This mapping was done through the `unicode-math` LuaLaTeX package, which keeps track of more than 2 400 mathematical symbols.

In addition, the LaTeX special symbols "{" and "}" were removed when not enclosing arguments passed to LaTeX commands or not defining sub/super script groups. Although these symbols can optionally be used to improve the readability of the mark-up language, unnecessary symbols can introduce inconsistencies and noise into the GT. Fig. 3 shows an example of this process.

As a final normalization step, all horizontal spacing commands were mapped to the `\hspace` command. This reduced the variability of the LaTeX transcriptions while maintaining the same visual representation of the MEs.

Validation of the reconstructed LaTeX code was done by comparing the original rendered ME with the rendering of the AST output, just before normalization. Error detection was done automatically by evaluating each pair of renderings with the Exact Match Image metric presented in Section 4.2.1. Approximately 5% of the

[9] https://www.ctan.org/pkg/nodetree?lang=en.

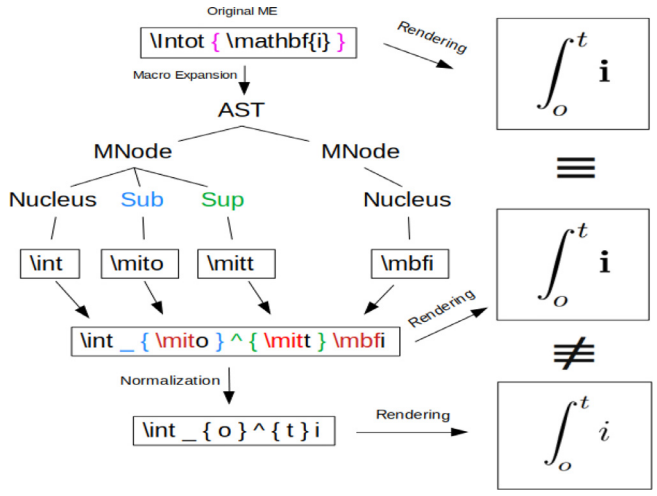


Fig. 3. Simplification of the normalization process, in which a LaTeX ME containing a user-defined macro is first converted to an AST, and then normalized. Normalization includes removing font information such as slant, style, and thickness (colored red), sub/super-script fixed order (colored blue and green), and flattening unnecessary groups (colored magenta).

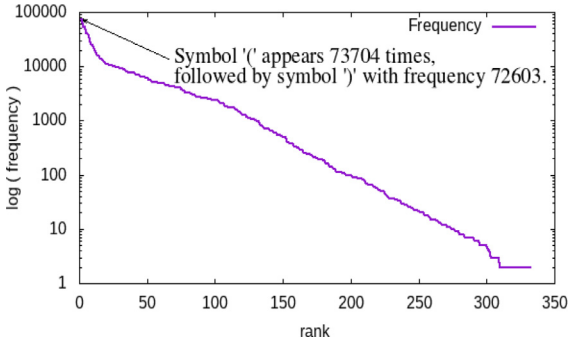


Fig. 4. A graph of the symbol rank versus frequency for the IBEM mathematical symbols.

reconstructed MEs had to be manually corrected to match the original rendering exactly.

3.4. Complementary dataset

As a by-product of the normalization process, a secondary dataset of isolated mathematical symbols was created. All individual images of the mathematical symbols were scaled and padded to a 28×28 pixels dimension, in grayscale, making abstraction of the font size. To group the different representations of the same symbol, the Unicode code points of the glyphs were mapped to a normalized LaTeX command, where font-specific information such as slant, style, and thickness was discarded. Each resulting class was then validated manually.

By processing the expressions found in the 600 documents presented in this paper, we obtained 1 109 926 glyphs, representing a total number of 332 different symbols. The classes include Latin and Greek letters, both upper and lower case, Arabic numerals, parentheses and brackets, and mathematical operators. These symbols are distributed as shown in Fig. 4. Classes with only one instance have been removed from the dataset as outliers. The frequency of the symbol "(" made up for 6, 64% of the total number of glyphs, closely followed by the symbol ")" with 6.54% (as expected). The next most frequent character was the digit "1" with a percentage of 4.89%.

Table 3
IBEM mathematical symbol recognition results for the *Model-C* system [20]. Experiments were performed under three scenarios, showing the effects of data-agumentation (Data Aug.) and minority class upsampling (Up-sample) on absolute error rate (AbsErr) and absolute class error rate (AbsClErr) metrics. The test comprises of 221 988 symbols in 332 different classes.

Exp.	Data Aug.	Up-sample	AbsErr	AbsClErr
1	No	No	25	17
2	Yes	No	27	3
3	No	Yes	1	1

4. Experiments and results

Several experiments can be carried out with the IBEM dataset GT: ME detection and extraction, symbol classification, ME recognition, etc. The dataset has already been used in an ICDAR 2021 ME detection competition [18]. The experiments reported in this paper focus on symbol classification and ME recognition, steps that take place once the MEs have been detected and extracted. These experiments aim to provide some benchmarks for comparison purposes so that future users of the IBEM dataset can have a baseline framework.

4.1. Symbol classification

This section presents benchmark experiments on classifying printed IBEM mathematical symbols. The prevalent technology for optical character recognition is based on Convolutional Neural Networks (CNN) to extract high-level features from images. Therefore, for this experiment, we chose a state-of-the-art classification system, *Model-C* [20]. This system is composed of 4 Convolutional blocks, followed by Batch Normalization, ReLU activation function and a 2×2 Max-Pooling layer. The authors also include Dropout layers for better generalization and a 1×1 Convolutional layer followed by a Global Average Pool layer to prune the parameters of the network. The classification of the symbols is done by using a Softmax layer.

As mentioned before, there are a total number of 1 109 926 glyphs in the IBEM dataset, grouped into 332 classes. We partitioned this data, assigning 80% for training and 20% for testing. Out of those training images, we chose a 20% subset for validation. The data was sampled in a stratified way to guarantee that the test set included all classes. The performance of the system was evaluated taking into account the absolute number of misclassified test samples (*AbsErr*) and the absolute number of classes completely misclassified (*AbsClErr*).

Table 3 shows the results of the classification system under three different scenarios that aim to address class imbalance through the use of data augmentation and minority class upsampling. The results of the first baseline experiment show that the system achieved very good classification results, with only 25 classification errors (*AbsErr*) out of a test set of 221 988 symbols. Considering that there are 38 classes with less than 5 instances, the systems misclassified only 17 classes (*AbsClErr*).

In the second experiment of Table 3, random rotations were applied in the range of $[-25, 25]$ degrees and zoom in the range of [0.9,1.1] to mimic slant and bold style glyphs and create more artificial data. When data augmentation was applied to generate 25% more training samples, the number of classes completely misclassified diminished significantly.

Lastly, the results of the third experiment show that through minority class upsampling the classification system achieved near-perfect results, even though no further data augmentation was used. In this last experiment, underrepresented classes were ran-

domly re-sampled from the training set to ensure that all classes have at least 5 instances (38 classes have less).

The results indicate that a CNN-based approach is suitable for extracting features from images of typeset mathematical symbols, and that, by data augmentation or upsampling of underrepresented classes, a feature extractor module could potentially be pre-trained as a precursor to a ME recognition system.

4.2. Mathematical expression recognition

4.2.1. ME Recognition systems and evaluation metrics

We provide benchmark results of ME recognition with the IBEM datasets using standard approaches. To facilitate the reproducibility of the experimental work, three open-source systems were used. The first one is based on structural and grammatical models (`Seshat`[10]) [9], while the second and third ones are based on deep neural networks with attention models (`Im2Latex`[11]) [7], and (`WAP`[12]) [10].

The `Seshat` ME recognition system is based on two-dimensional probabilistic context-free grammars (2D-PCFG). In this system, the connected components of the image pixels are computed, which are then structurally merged using productions of the 2D-PCFG. This process is recursively applied with the Viterbi algorithm that allows to structural combine large portions of the ME and accounts for long-term dependencies. The spatial and geometric information is used to constrain the search space to make the search feasible [9]. The hyperparameters of the model have been fitted on a subset of the training set (validation set) with the downhill simplex algorithm [21], minimizing the LaTeX token error rate. A noticeable feature of this system is that it is known how to get not only the 1-best interpretation but a hypergraph with thousands of possible interpretations [22]. Another important characteristic of syntax-based systems is that it is well-known how to define syntactic relations among different parts of the ME.

The `Im2Latex` ME recognition system is based on deep neural networks, in which the sequence-to-sequence attention-based encoder-decoder architecture is employed to tackle the image-to-mark-up conversion problem. In this system, the image features of the MEs are encoded through a multilayer CNN and a bidirectional Long Short-Term Memory (LSTM). The LaTeX mark-up language of the MEs is generated through an LSTM attention-based decoder.

Lastly, the `WAP` ME recognition system, similar to the `Im2Latex` system, is based on deep neural networks. Although this system also employs an attention-based encoder-decoder framework, it features a deep fully convolutional architecture as an encoder. For the recognition experiments of this paper, we employed the system presented in [10], which is an improved version of the original [23] system, with changes to both the encoder and the attention mechanism.

The `Seshat` system is based on a handcrafted 2D-PCFG that was not adapted to the IBEM dataset, and therefore not all IBEM MEs could be parsed. Although Table 4 also reports the data under the constraints of the `Seshat` system, note that all results are reported without considering this restriction.

To better understand the difficulty of the ME recognition task, Fig. 5 shows the ME distribution as a function of the number of mark-up LaTeX words. As can be seen, the inline MEs are much shorter than the displayed MEs and have significantly more sharpened distributions. In addition, approximately 20% of displayed MEs are multi-line, as opposed to less than 2% of inline ME. More than 19% of displayed MEs contain alignment symbols (denoting arrays, matrices, or vertically aligned mathematical subexpressions), as opposed to less than 0.1% for inline MEs. Furthermore, approximately 45% of displayed ME contain fractions (complex two-dimensional elements), as opposed to approximately 2% of inline MEs. Finally, it is important to mention that, although near-perfect results are obtained for symbol classification, ME recognition also requires symbol segmentation (as a prerequisite or computed on the fly) and structure analysis. These characteristics are fundamental for interpreting the results obtained by the recognition systems.

Table 4
Number of MEs in the training and test partitioning of the IBEM dataset, showing the difference between the original data and the data that could be parsed by the 2D-PCFG.

Filter	ME type	Training	Test	Total
Original	Displayed	21 867	7 736	29 603
	Inline	100 442	36 647	137 089
	Total	122 309	44 383	**166 692**
2D-PCFG Filter	Displayed	17 791	7 736	25 527
	Inline	97 677	36 647	134 324
	Total	115 468	44 383	**159 851**

The evaluation of the ME recognition systems was performed with the following metrics:

- **Exact Match Mark-up** (*ExM* $\leq \epsilon$): LaTeX mark-up language level metric that measures the percentage of 1-best hypotheses that match the associated ground truth ME when up to and including ϵ structural or symbol errors can be tolerated, where $\epsilon \in \{0, 1, 2\}$ (See expression recognition rate [24]). Note that if $\epsilon = 0$ a perfect symbol and structure recognition is required. Thus, this metric is sensitive to the length of the MEs and tends to be overly pessimistic in the case of larger MEs.
- **Exact Match Image** (*ExIm*): image level metric that measures the percentage of rendered hypotheses that match the corresponding rendered references. Both references and hypotheses were rendered as standalone MEs to eliminate unwanted noise caused by the surrounding context. Image pairs are considered to match exactly if they display a misalignment of less than 5 pixels [25].
- **Bleu**: (*Bleu*) score [26] measures the similarity between the model's best prediction and the associated ME reference in accordance with n-gram accuracy. Scores up to 4-grams were considered for this metric, taking into account the size of the MEs.

4.2.2. Results

Table 5 shows the main experimental results on the IBEM ME recognition task. It can be seen that the `WAP` and `Im2Latex` systems significantly outperform the `Seshat` system. In the case of inline MEs, all systems demonstrate significantly improved recognition results compared to displayed MEs. This can be attributed to the shorter length and generally less complex structure of inline MEs.

While all systems generally perform well when evaluated with *ExIm* and *Bleu*, with *ExM*, the results are considerably worse. *ExM* is a very strict metric since it requires both a perfect symbol and structure recognition. By softening this restriction and allowing up to two structural or symbol recognition errors (ExM$\leq \epsilon, \epsilon \in \{1, 2\}$), it becomes apparent that all systems generate a substantial proportion of hypotheses that closely align with their corresponding references. However, it is crucial to consider that, for displayed MEs, allowing a recognition tolerance of up to two symbol or structural

[10] https://www.github.com/falvaro/seshat.
[11] https://www.github.com/shchae7/im2latex.
[12] https://www.github.com/jmwang66/WAP-implemented-by-Pytorch.

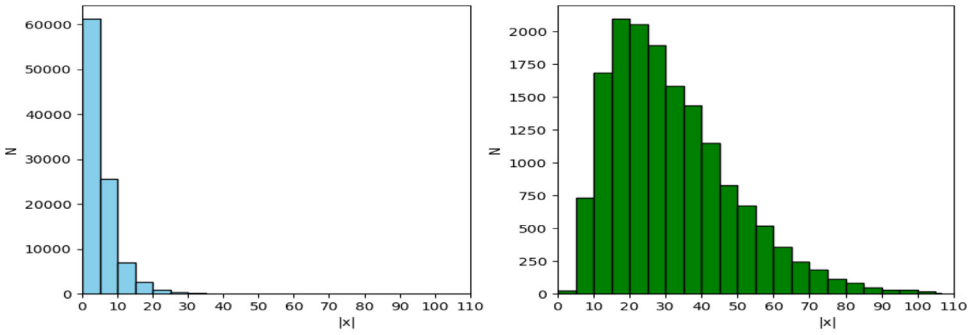


Fig. 5. Histogram representing the distribution of the ME based on the number of mark-up LaTeX words/symbols defined in the MEs.

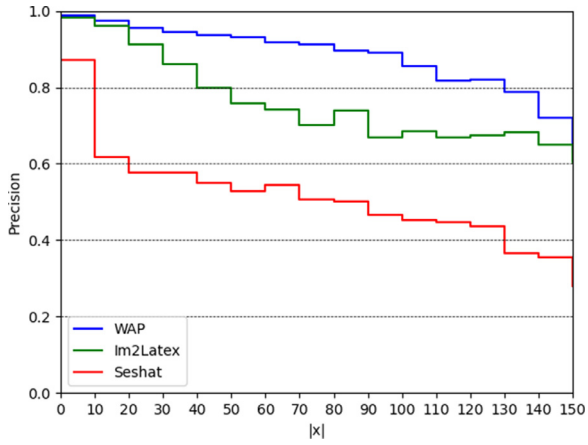


Fig. 6. Precision of *Bleu* interpolated score, up to 4-grams, based on ME size. The MEs are grouped in consecutive intervals of size 10. Due to low representation, MEs with more than 150 symbols have been excluded from this chart, although they are included in the results of Table 5.

errors is justifiable. In contrast, for inline MEs, which are significantly shorter, applying this adjustment could profoundly alter the semantics of the recognized expressions. This factor must be considered when analyzing the performance of recognition models for inline MEs.

To highlight the potential semantic ambiguity of the IBEM dataset, all systems are also evaluated with the *ExIm* metric. As seen in Table 5, all recognition systems achieve a much better visual match when compared to the more restrictive textual match. This implies that the language models of all systems have learned alternative ways of transcribing a ME while keeping the same visual representation. This feature could prove beneficial for ME retrieval approaches that are image-based, given that different textual representations of the same rendered ME could be more likely paired. However, please note that symbols such as `\sum` and `\Sigma` have the same visual representation, with the semantic consequences it implies.

In light of these results, Fig. 6 shows the *Bleu* score precision of these systems as a function of the number of mark-ups LaTeX words/symbols make up the MEs. It can be seen that the attention mechanisms of the neural-based systems learn to mitigate the effect of having to recognize long MEs well.

Table 6 shows results based on the structure of the MEs. The performance of all systems drops significantly for multiline (*Mult.*) MEs or for MEs that contain alignment symbols (*Align.*). These two elements are highly connected, as aligned components such as matrices are always multiline, and multiline MEs feature subexpressions that are usually vertically aligned. The task of correctly recognizing MEs of these characteristics is particularly difficult as the relationship between symbols is greatly extended.

Table 5
ME baseline recognition results obtained on the IBEM test set. The Seshat [9], Im2Latex [7] and WAP [10] systems are evaluated on the $ExM{\leq}\epsilon$, *ExIm* and *Bleu* metrics with scores between 0 and 100.

ME	Model	ExM=0	ExM≤1	ExM≤2	ExIm	Bleu
Disp.	Seshat	6.0	6.7	7.7	17.1	52.9
	Im2Latex	25.4	32.2	36.7	42.8	73.9
	WAP	42.8	53.0	61.6	64.0	86.0
Inline	Seshat	70.4	72.1	74.9	81.8	81.0
	Im2Latex	93.7	95.8	96.7	99.3	97.4
	WAP	95.7	97.9	98.6	99.1	98.3
All	Seshat	59.2	61.3	63.8	70.5	76.1
	Im2Latex	81.8	84.6	86.1	89.5	93.3
	WAP	**86.5**	**90.0**	**92.1**	**93.0**	**96.2**

Table 6
ME baseline recognition results based on the type of components contained in the MEs. Please note that results are reported at ME level and not at component level, as the latter would require using a LaTeX ME parser. In the case of *Seshat*, multiline or MEs containing alignment symbols could not be parsed by the system.

ME	Model	ExM=0	ExM≤1	ExM≤2	ExIm	Bleu
Mult.	Seshat	-	-	-	-	-
	Im2Latex	0.4	0.9	1.4	1.0	47.2
	WAP	5.2	7.8	12.9	8.6	55.8
Align.	Seshat	-	-	-	-	-
	Im2Latex	1.0	1.5	1.9	3.5	49.3
	WAP	6.2	8.8	14.2	11.9	57.1
Frac.	Seshat	6.5	7.7	10.4	16.0	14.6
	Im2Latex	31.1	38.4	43.6	47.7	81.3
	WAP	**54.4**	**65.9**	**75.4**	**75.8**	**94.9**
Rest	Seshat	65.0	66.4	69.0	76.2	74.1
	Im2Latex	89.6	92.1	93.3	95.1	96.2
	WAP	**92.7**	**95.6**	**96.9**	**97.6**	**97.9**

In the case of MEs containing fractions (*Frac.*), for which multiline or aligned MEs were not considered, the `WAP` system performs exceptionally well, showing that this system is better suited for recognizing complex structures, as is also evident in Fig. 6. Lastly, Table 6 shows that for the rest of MEs that are not multiline or do not contain alignment symbols or fractions, all systems achieve very high scores for all metrics, indicating that the main recognition challenges of the IBEM corpus are the correct interpretation of multiline, matrices and vertically aligned MEs, and to a lesser degree, MEs containing fractions.

Finally, the recognition systems are evaluated on test images automatically detected and extracted. These images result from the ICDAR 2021 Competition on Mathematical Formula Detection [18], in which participants were asked to automatically detect MEs on

Table 7
Results of ME recognition with automatic ME detection methods, based on the coordinates submitted by the winner [19] of the Mathematical Formula Detection Competition [18].

ME	Model	ExM=0	ExM≤1	ExM≤2	ExIm	Bleu
	Seshat	4.0	4.9	6.6	16.8	45.9
Disp.	Im2Latex	14.5	19.8	25.2	26.7	63.6
	WAP	28.5	39.1	48.4	52.7	80.5
	Seshat	58.8	61.5	64.1	70.7	69.2
Inline	Im2Latex	65.5	77.9	81.4	78.8	78.2
	WAP	74.1	81.5	86.8	87.7	85.1
	Seshat	49.2	51.6	53.9	61.3	65.1
All	Im2Latex	56.6	67.8	71.7	69.7	75.6
	WAP	**66.2**	**73.9**	**79.9**	**81.6**	**84.3**

two subsets of the test set. An image was considered as correctly detected if the predicted and ground-truth expressions overlapped with an Intersection-over-Union (IoU) threshold of ≥ 0.7. Table 7 shows the evaluation of the recognition systems, on the basis of the test images coordinates submitted by the winner [19] of the competition. It is important to mention that the ME recall of this detection system was of $95,47\%$. As can be observed, all recognition systems are affected by automatic ME detection, where symbol clipping introduces never-before-seen glyph samples. This particularly affects inline MEs, as they generally have fewer symbols, and a 70% IoU overlap leads to a higher loss of information.

The end objective of the IBEM dataset is to facilitate the indexing and searching of MEs in massive collections of STEM documents. For ME retrieval tasks, it is reasonable to assume that user-defined queries are mathematical sub-expressions of reduced size. While, overall, the neural-based systems outperformed the `Seshat` system, it is essential to remark that structural and grammatical models provide not only the recognition results but also the parse tree, actually the n-best parse trees. Therefore, these models are more suitable for ME retrieval, as MEs are broken down into sub-expressions providing the relationship between them [8]. This is an important problem for which grammatical models offer complete solutions.

5. Conclusion

This paper introduces the IBEM dataset for researching printed mathematical expression recognition and searching. The dataset includes a rich GT that consists of the LaTeX transcript and the position of each ME in the rendered images. The IBEM dataset has been prepared automatically, and it currently has 600 publicly available documents, with more than 8 200 page images and more than 160 000 mathematical expressions. The proposed method is scalable, and we expect to scale up the dataset significantly.

Along with the IBEM dataset, we provide baseline experiments for both symbol classification and ME recognition. These experiments aim to provide some benchmarks that may be useful to future users of the IBEM dataset. The corpus presented in this paper is available on the ZENODO platform.[13]

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The manuscript includes links to the data used for the research described in the article.

[13] https://www.zenodo.org/record/7963703.

Acknowledgments

This work has been partially supported by MCIN/AEI/10.13039/501100011033 under the grant PID2020-116813RB-I00; the Generalitat Valenciana under the FPI grant CIACIF/2021/313; and by the support of the Valencian Graduate School and Research Network of Artificial Intelligence.

References

[1] R. Zanibbi, D. Oard, A. Agarwal, B. Mansouri, Overview of ARQMath 2020: CLEF Lab on answer retrieval for questions on math, in: LNCS, 2020, pp. 169–193.
[2] R. Zanibbi, D. Blostein, Recognition and retrieval of mathematical expressions, IJDAR 14 (2011) 331–357.
[3] W. Ohyama, M. Suzuki, S. Uchida, Detecting mathematical expressions in scientific document images using a u-net trained on a diverse dataset, IEEE Access 7 (2019) 144030–144042.
[4] M. Mahdavi, R. Zanibbi, H. Mouchère, C. Viard-Gaudin, U. Garain, ICDAR 2019 CROHME + TFD: Competition on recognition of handwritten mathematical expressions and typeset formula detection, in: ICDAR, 2019, pp. 1533–1538.
[5] B. Hai Phong, L. Dat, N. Yen, T. Hoang, T. Le, A deep learning based system for mathematical expression detection and recognition in document images, in: KSE, 2020, pp. 85–90, doi:10.1109/KSE50997.2020.9287693.
[6] X. Lin, L. Gao, Z. Tang, J. Baker, V. Sorge, Mathematical formula identification and performance evaluation in PDF documents, IJDAR 17 (2013) 239–255.
[7] Y. Deng, A. Kanervisto, A. Rush, What you get is what you see: avisual markup decompiler, ArXiv abs/1609.04938 (2016).
[8] E. Noya, J. Benedí, J. Sánchez, D. Anitei, Discriminative learning of two-dimensional probabilistic context-free grammars for mathematical expression recognition and retrieval, in: IbPRIA, 2022, pp. 333–347.
[9] F. Álvaro, J. Sánchez, J. Benedí, An integrated grammar-based approach for mathematical expression recognition, Pattern Recognit. 51 (2016) 135–147.
[10] J. Zhang, J. Du, L. Dai, Multi-scale attention with dense encoder for handwritten mathematical expression recognition, in: ICPR, 2018, pp. 2245–2250.
[11] J. Gehrke, P. Ginsparg, J. Kleinberg, Overview of the 2003 KDD cup, SIGKDD Explor. Newsl. 5 (2) (2003) 149–151.
[12] I. Phillips, Methodologies for using UW databases for OCR and image understanding systems, in: SPIE, volume 3305, 1998, pp. 112–127.
[13] F. Shafait, J. van Beusekom, D. Keysers, T. Breuel, Page frame detection for marginal noise removal from scanned documents, in: Image Analysis, 2007, pp. 651–660.
[14] M. Suzuki, S. Uchida, A. Nomura, A ground-truthed mathematical character and symbol image database, in: ICDAR, 2005, pp. 675–679.
[15] S. Zhu, L. Hu, R. Zanibbi, Rotation-robust math symbol recognition and retrieval using outer contours and image subsampling, SPIE 8658 (2013) 05, doi:10.1117/12.2008383.
[16] F. Álvaro, J. Sánchez, Comparing several techniques for offline recognition of printed mathematical symbols, in: ICPR, 2010, pp. 1953–1956.
[17] M. Suzuki, F. Tamari, R. Fukuda, S. Uchida, T. Kanahori, Infty- an integrated OCR system for mathematical documents, in: ACM, 2003, pp. 95–104.
[18] D. Anitei, J. Sánchez, J. Fuentes, R. Paredes, J. Benedí, ICDAR 2021 competition on mathematical formula detection, in: ICDAR, 2021, pp. 783–795.
[19] Y. Zhong, X. Qi, S. Li, D. Gu, Y. Chen, P. Ning, R. Xiao, 1St place solution for ICDAR 2021 competition on mathematical formula detection, ArXiv abs/2107.05534 (2021).
[20] L. Dong, H. Liu, Recognition of offline handwritten mathematical symbols using convolutional neural networks, in: Image and Graphics, 2017, pp. 149–161.
[21] J. Nelder, R. Mead, A simplex method for function minimization, Comput. J. 7 (4) (1965) 308–313.
[22] E. Noya, J. Sánchez, J. Benedí, Generation of hypergraphs from the n-best parsing of 2D-probabilistic context-free grammars for mathematical expression recognition, in: ICPR, 2021, pp. 5696–5703.
[23] J. Zhang, J. Du, S. Zhang, D. Liu, Y. Hu, J. Hu, S. Wei, L. Dai, Watch, attend and parse: an end-to-end neural network based approach to handwritten mathematical expression recognition, Pattern Recognit. 71 (2017) 196–206.
[24] H. Mouchère, C. Viard-Gaudin, R. Zanibbi, U. Garain, ICFHR 2014 competition on recognition of on-line handwritten mathematical expressions (CROHME 2014), in: ICFHR, 2014, pp. 791–796.
[25] Y. Deng, A. Kanervisto, J. Ling, A.M. Rush, Image-to-markup generation with coarse-to-fine attention, in: ICML, 2017, pp. 980–989.
[26] K. Papineni, S. Roukos, T. Ward, W. Zhu, BLEU: a method for automatic evaluation of machine translation, in: ACL, 2002, pp. 311–318.