Pattern Analysis and Applications (2023) 26:1571–1584
https://doi.org/10.1007/s10044-023-01158-8

ORIGINAL ARTICLE

# Discriminative estimation of probabilistic context-free grammars for mathematical expression recognition and retrieval

**Ernesto Noya**[1] · **José Miguel Benedí**[1,2] · **Joan Andreu Sánchez**[1] · **Dan Anitei**[1]

Received: 30 August 2022 / Accepted: 21 March 2023 / Published online: 18 April 2023


**Abstract**
We present a discriminative learning algorithm for the probabilistic estimation of two-dimensional probabilistic context-free grammars (2D-PCFG) for mathematical expressions recognition and retrieval. This algorithm is based on a generalization of the H-criterion as the objective function and the growth transformations as the optimization method. For the development of the discriminative estimation algorithm, the *N*-best interpretations provided by the 2D-PCFG have been considered. Experimental results are reported on two available datasets: *Im2Latex* and *IBEM*. The first experiment compares the proposed discriminative estimation method with the classic Viterbi-based estimation method. The second one studies the performance of the estimated models depending on the length of the mathematical expressions and the number of admissible errors in the metric used.

**Keywords** Discriminative learning · Two-dimensional probabilistic context-free grammars · Mathematical expression retrieval · Probabilistic indexing

## 1 Introduction

Syntactic models have been demonstrated to be a fundamental formalism for pattern recognition since they introduce effective restrictions in the solution search space for structured interpretation problems. Thus, finite-state language models provide a prior probability in many current applications, like automatic speech recognition (ASR) [1], machine translation (MT) [2], and handwritten text recognition (HTR) [3], that makes the decoding problem feasible. A noticeable characteristic of syntactic models is that they can provide efficiently a possible interpretation for a given input or a sorted set of *N*-best alternative interpretations for the same input. The computation of *N*-best solutions for stochastic finite-state models [4] and probabilistic context-free grammars (PCFG) has been studied in the past [5]. These *N*-best solutions can be represented as a word graph [6] or a hypergraph [7] that can generalize and provide alternative hypotheses not previously included in the *N*-best solutions.

The word graphs obtained from stochastic finite-state models are meaningful representations used in ASR [6], MT [8], and HTR [9] since they can obtain confidence measures at the word or sentence levels. Hypergraphs computed from PCFG can be used for the same purpose, and they have been used in the past for interactive parsing [10] and mathematical expression recognition [7]. Probabilistic training from data is usually performed by optimizing a goal function and using some statistical optimization framework. It is of paramount importance to take profit as much as possible of the data in the case of a limited amount of training samples. Probabilistic training of PCFG has been researched using the maximum likelihood criterion and optimizing this function in the optimization framework based on growth transformations [11].

PCFGs are a powerful formalism for parsing mathematical expressions (ME) since they are suitable for capturing long-term dependencies among the different elements in a

✉ Joan Andreu Sánchez
jandreu@prhlt.upv.es

Ernesto Noya
ernogar@prhlt.upv.es

José Miguel Benedí
jmbenedi@prhlt.upv.es

Dan Anitei
danitei@prhlt.upv.es

[1] Universitat Politècnica de València, Valencia, Spain

[2] valgrAI: Valencian Graduate School and Research Network of Artificial Intelligence, València, Spain

ME and hierarchical dependencies on large regions of the ME. This paper considers a two-dimensional extension of PCFG (2D-PCFG) [12, 13] that deals with the inherent spatial ambiguity of ME. The syntactic rules of a 2D-PCFG can be defined manually in a heuristic way. However, the probabilities of the rules can fit the task and be learned automatically. There are few datasets for training 2D-PCFG [14]. Recently, large datasets have been made publicly available that can be used for training the models [15]. However, even with these datasets, more training data are needed to represent the considerable variability in ME. Additionally, more efficient methods that take more profit from the training data have to be devised. This paper researches the use of discriminative techniques for the probabilistic estimation of 2D-PCFG [16, 17]. This approach combines correct and incorrect interpretations in a discriminative way [18], in contrast to classical estimation methods, where incorrect interpretations are not explicitly used. This usually allows us to get better results and speeds up the training process.

Current results on ME recognition are far from being perfect for searching purposes in large image datasets, and having accurate enough results may be impossible given the intrinsic ambiguity present in ME. Therefore, more flexible approaches for searching ME have to be devised. Similar problems have been researched for HTR [9], and the adopted solution is based on getting an adaptive list of hypotheses for each word image obtained from word graphs. This solution is based on probabilistic indexing (PrIx) [9]. In this paper, we intend to follow a similar PrIx research approach for ME searching. In this sense, it is relevant to consider the types of queries that users could make for searching ME. The concept of "word" is not defined for ME, and we assume in this paper that users may be interested in searching for the whole ME and subexpressions contained in that whole expression. Therefore, evaluating the recognition system at the whole ME level and a sub-expression level is essential.

This paper extends [19] by introducing new demonstrations of the theoretical fundamentals of the discriminative probabilistic estimation algorithm introduced in Sect. 3 and by presenting more comprehensive and consolidated experiments in Sect. 5.

## 2 Related work

Past and recent surveys from 2011 [20] to 2020 [21] and competitions of ME recognition and retrieval like CROHME [22] and OffRaSHME [23] show most teams divide the problem in several steps, namely, (1) symbols segmentation and recognition where convolutional and recurrent neural networks (CRNN) are used for online strokes or convolutional neural networks (CNN) for offline symbols [24], (2) layout analysis to determine spatial relationships, and (3) syntactical analysis of the structure with grammars, trees, or rule-based approaches [20]. Many teams have recently tried end-to-end solutions that can generate LaTeX directly from the image, merging all steps in an CRNN or bidirectional long short-term memory (BLSTM) NN [21], and introducing new architectures like adversarial networks [25], graph networks [26], or transformers [27].

One disadvantage of most end-to-end NN is that they do not provide solutions to generate the syntactic tree structure that is inherent to the ME. Given the ambiguity within ME and the language used to describe them like LaTeX, the retrieval problem greatly benefits from knowing the syntactic (and semantic) structure of the expression. For this reason, teams interested in recognition as a previous step to search usually try different types of graphs [28] or other structures [29] to compare between MEs. A study done to measure the effectiveness of ME search solutions shows that users search for mathematical information differently than textual searches, and current solutions still need to be investigated before being used [30].

2D-PCFGs have been proposed for ME recognition [13] since they allow the generation of a hierarchical structure that accounts for mathematical symbols and relationships among different parts of a ME. The inherent ambiguity associated with the ME recognition process can be overcome by considering a set of $N$-best parse trees if 2D-PCFGs are used. Regarding the probabilistic estimation of PCFG, a previous work [7] has studied this problem using the likelihood function as the merit function to be optimized. However, a recent research has demonstrated that better results can be achieved by considering a discriminative function as a merit function [18].

## 3 Problem formulation and notation

The input domain in the printed mathematical expression recognition and retrieval is the set of images or regions of an image that can contain a ME. Given an input image, the first step is to define a representation function that maps the image to another representation more suitable for solving the problem. The selected representation for printed document images is usually based on connected components. Figure 1 shows an input ME and its representation in terms of connected components.

As shown in Fig. 1, this ME consists of 11 connected components, most of which represent a single symbol of the ME. However, there are also symbols formed by more than one connected component (e.g., $i$ and $=$). Furthermore, the connected components alone are insufficient to address the problem of ME recognition. For instance, the connected components associated with the symbols $x$ and 2 do not explain by themselves the relationship between

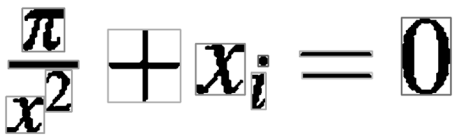

**Fig. 1** Input image example for the math expression $\frac{\pi}{x^2} + x_i = 0$ and its set of associated connected components

them to define the subexpression $x^2$. Therefore, the ultimate purpose should not only be the recognition of the symbols but should also include the recognition of the structure that relates them. For this reason, many approaches to ME recognition and retrieval were based on probabilistic grammar models [13, 31, 32] because they constitute a natural way to model this kind of problems.

We pose the ME recognition and retrieval as a structural parsing problem, in which the main goal is to obtain the set of symbols and the structure that defines their relationships from an input image. Formally, let $\boldsymbol{x} = \{x_1, \ldots, x_{|\boldsymbol{x}|}\}$ be a set of connected components from an input ME to be recognized or as a search query, where $|\boldsymbol{x}|$ is the number of connected components associated with the input image. The aim is to obtain the most likely sequence of mathematical symbols $\boldsymbol{s} \in \mathcal{S}$ related among them according to the most likely syntactic parse $\boldsymbol{t} \in \mathcal{T}$ given $\boldsymbol{x}$. $\mathcal{S}$ is the set of all possible sequences of (pre-terminal) symbols, and $\mathcal{T}$ represents the set of all possible syntactic parses, such that $\boldsymbol{s} = \text{yield}(\boldsymbol{t})$. We can describe it as follows:

$$(\hat{\boldsymbol{t}}, \hat{\boldsymbol{s}}) \approx \underset{\substack{\boldsymbol{t} \in \mathcal{T}, \boldsymbol{s} \in \mathcal{S} \\ \boldsymbol{s} = \text{yield}(\boldsymbol{t})}}{\arg\max} p(\boldsymbol{t}, \boldsymbol{s} \mid \boldsymbol{x}) \approx \underset{\substack{\boldsymbol{t} \in \mathcal{T}, \boldsymbol{s} \in \mathcal{S} \\ \boldsymbol{s} = \text{yield}(\boldsymbol{t})}}{\arg\max} p(\boldsymbol{s} \mid \boldsymbol{x})\, p(\boldsymbol{t} \mid \boldsymbol{s}) \tag{1}$$

$p(\boldsymbol{s} \mid \boldsymbol{x})$ represents the observation (symbol) likelihood and $p(\boldsymbol{t} \mid \boldsymbol{s})$ represents the structural probability. We consider Eq. (1) a holistic search problem, where symbol segmentation, symbol recognition, and the structural analysis of the input expression are globally achieved [13].

In this paper, we will focus on the parsing problem associated with the computation of the structural probability $p(\boldsymbol{t} \mid \boldsymbol{s})$ and especially on estimating the grammatical models used to tackle Eq. (1). We first introduce the notation used in this paper.

**Definition 1** A context-free grammar (CFG), $G = (\mathcal{N}, \Sigma, S, \mathcal{P})$, is a tuple where $\mathcal{N}$ is a finite set of non-terminal symbols, $\Sigma$ is a finite set of terminal symbols ($\mathcal{N} \cap \Sigma = \emptyset$), $S \in \mathcal{N}$ is the start symbol of the grammar, and $\mathcal{P}$ is a finite set of rules: $A \to \alpha$, where $A \in \mathcal{N}$ and $\alpha \in (\mathcal{N} \cup \Sigma)^+$.

A CFG in Chomsky Normal Form (CNF) is a CFG in which the rules are of the form $A \to BC$ or $A \to a$, where $A, B, C \in \mathcal{N}$ and $a \in \Sigma$.

**Definition 2** A probabilistic CFG (PCFG) is defined as a pair $(G, p)$, where $G$ is a CFG and $p : P \to ]0, 1]$ is a probability function of rule application such that $\forall A \in \mathcal{N} : \sum_{i=1}^{n_A} p(A \to \alpha_i) = 1$, where $n_A$ is the number of rules associated with non-terminal symbol $A$.

**Definition 3** A two-dimensional PCFG (2D-PCFG), $\mathbb{G}$, is a generalization of a PCFG, where terminal and non-terminal symbols describe bi-dimensional regions. This grammar in CNF results in two types of rules: terminal and binary rules.

The terminal rules, $A \to a$, represent the mathematical symbols that are ultimately the terminal symbols of 2D-PCFG. The probability $p(A \to a)$, therefore, depicts the probability that $A$ is the solution to the elementary problem $a$. The binary rules, $A \xrightarrow{r} BC$, have an additional parameter ($r$) representing the given spatial relationship between the $B$ and $C$ subproblems. Moreover, this means that $A$ is the solution to the subproblems associated with $B$ and $C$ regions compatible with the spatial relationship $r$. This work considers six spatial relationships: right, below, subscript, superscript, inside, and nth root [13]. The probability of the binary rule, $A \xrightarrow{r} BC$, is defined by

$$(A \xrightarrow{r} BC) \overset{\text{def}}{=} p(BC, r|A) \approx p(BC|A)\ p(r|BC)$$

$p(BC \mid A)$ is the probability of the binary rule of a PCFG, and $p(r \mid B, C)$ is the probability that regions encoded by non-terminals $B$ and $C$ are arranged according to the spatial relationship $r$.

Let $\mathbb{G}$ be a 2D-PCFG and let $\boldsymbol{x}$ be a set of connected components. We denote $\mathcal{T}_x$ as the set of all possible parse trees for $\boldsymbol{x}$. The expression $\text{N}(A \to \alpha, t_x)$[1] represents the number of times that the rule $A \to \alpha$ has been used in the parse tree $t_x \in \mathcal{T}_x$, and $\text{N}(A, t_x)$ is the number of times that the non-terminal $A$ has been used in $t_x$. It should satisfy that $\text{N}(A, t_x) = \sum_{i=1}^{n_A} \text{N}(A \to \alpha_i, t_x)$. With all that, we define the following expressions:

- *Probability of the parse tree* $t_x$ of $\boldsymbol{x}$

$$P(\boldsymbol{x}, t_x) = \prod_{\forall (A \to \alpha) \in \mathcal{P}} p(A \to \alpha)^{\text{N}(A \to \alpha, t_x)}$$

- *Probability* of $\boldsymbol{x}$

[1] To reduce the complexity of the notation, we will henceforth use $(A \to \alpha)$ instead of $(A \xrightarrow{r} \alpha)$.

$$P(\boldsymbol{x}) = \sum_{\forall t_x \in \mathcal{T}_x} P(\boldsymbol{x}, t_x) \tag{2}$$

- *Probability of the best parse tree* of $\boldsymbol{x}$

$$\widehat{P}(\boldsymbol{x}) = \max_{\forall t_x \in \mathcal{T}_x} P(\boldsymbol{x}, t_x) \tag{3}$$

- *Best parse tree* of $\boldsymbol{x}$

$$\widehat{t}_x = \arg\max_{\forall t_x \in \mathcal{T}_x} P(\boldsymbol{x}, t_x)$$

Expressions (2) and (3) can be calculated, respectively, using modified versions of the well-known *inside* [33] and *Viterbi* [34] algorithms for 2D-PCFG [7]. We can also calculate the *N*-best parse trees for 2D-PCFG [7]. Furthermore, given $\Delta_x \subseteq \mathcal{T}_x$, a finite subset of derivations of $\boldsymbol{x}$, we can also define:

- Probability of $\boldsymbol{x}$ with respect to $\Delta_x$

$$P(\boldsymbol{x}, \Delta_x) = \sum_{\forall t_x \in \Delta_x} P(\boldsymbol{x}, t_x) \tag{4}$$

- Probability of the best parse tree of $\boldsymbol{x}$ with respect to $\Delta_x$

$$\widehat{P}(\boldsymbol{x}, \Delta_x) = \max_{\forall t_x \in \Delta_x} P(\boldsymbol{x}, t_x) \tag{5}$$

These expressions, respectively, coincide with expressions (2) and (3) when $\Delta_x = \mathcal{T}_x$.

## 4 Discriminative learning of 2D-PCFGs

Given a representative training sample $\Omega$ and a particular model defined by its set of parameters $\theta$, the problem of estimating the parameters $\theta$ of the model can state as follows:

$$\widehat{\theta} = \arg\max_{\theta} f_\theta(\Omega)$$

where $f_\theta(.)$ is the *objective function* to be optimized. Two issues should be considered: the optimization method and the objective function selection. In this paper, we consider an optimization method based on the *growth transformation* (GT) framework [35, 36] and an objective function derived from a generalization of the H-criterion [17, 37].

In this section, we present the original H-criterion and then propose the generalized H-criterion as an objective function for discriminative training of a 2D-PCFG. Next, we develop the method of growth transformations for the generalized H-criterion and define some related discriminative algorithms.

### 4.1 H-criterion

The H-criterion-based learning framework was proposed by Gopalakrishnan et al. in [37] as a generalization of the estimators of maximum likelihood (ML), maximum mutual information (MMI), and conditional maximum likelihood (CML). We can state it in this way: let $\Omega = \{(x_i, y_i)\}_{i=1}^N$ be the training sample, where $x_i$ are the input observations, and $y_i$ are the reference interpretations; and let $\theta$ be the model's parameters to be estimated. An H-estimator, $\widehat{\theta}(a, b, c)$, can be obtained by minimizing the H-criterion as follows:

$$H_{a,b,c}(\theta; \Omega) = -\frac{1}{n} \sum_{i=1}^{n} \log\, p_\theta^a(x_i, y_i)\; p_\theta^b(x_i)\; p_\theta^c(y_i) \tag{6}$$

Parameters $a, b$ and $c$ are constants, and $a > 0$ is fulfilled. Therefore, the ML estimator can be represented by $\widehat{\theta}(1, 0, 0)$, the MMI estimator by $\widehat{\theta}(1, -1, -1)$, and the CML estimator by $\widehat{\theta}(1, 0, -1)$ [37].

### 4.2 Generalized H-criterion for 2D-PCFGs

In this section, we propose a new criterion function for the discriminative estimation of parameters (probabilities of the rules) of a 2D-PCFG. This new function is a generalization of the H-criterion mentioned above [17, 37].

Given a training sample $\Omega = \{(x_i, t_i)\}_{i=1}^N$, where $x_i$ are the input sequences of connected components and $t_i$ are the reference parse trees, a 2D-PCFG, $\mathbb{G}$ (Def. 3), where the model parameters to be estimated are the probabilities of the terminal and binary rules, and a set of parse trees $\Delta_x$, obtained by a parsing process from the $\mathbb{G}$ model, we propose a new method of estimating the parameters of $\mathbb{G}$ using the generalized H-criterion by minimizing the following expression (see Eq. (6)):

$$H_{1,-h,0}(\mathbb{G}, \Omega) = -\frac{1}{|\Omega|} \log \widetilde{F}_h(\mathbb{G}, \Omega) = -\frac{1}{|\Omega|} \log \prod_{x \in \Omega} \frac{P(x, \Delta_x^r)}{P(x, \Delta_x^c)^h} \tag{7}$$

where $0 \leq h < 1$ and $\Delta_x^r \subset \Delta_x^c$ must be fulfilled. The set $\Delta_x^r$ must contain only the correct parse trees of sentence $x$. In contrast, the set $\Delta_x^c$ must contain competing parse trees of sentence $x$. If $h > 0$, then the generalized H-criterion can be viewed as a discriminative learning method. The exponent $h$ aims to establish the degree to which the competing parse trees discriminate against the correct parse trees. Optimizing the generalized H-criterion attempts simultaneously to maximize the numerator term $P(x, \Delta_x^r)$ and to minimize the denominator term $P(x, \Delta_x^c)^h$ for each observation $x \in \Omega$ of the training sample.

### 4.3 Growth transformations for generalized H-criterion

The objective function, obtained from the generalization of the H-criterion according to Eq. (7), can be optimized using growth transformations for rational functions. The growth transformations were initially developed for a homogeneous polynomial with positive coefficients [35]. Gopalakrishnan et al. extended in [16] this result to optimize rational functions. Since $\widetilde{F}_h(\cdot,\cdot)$ is a rational function (see Eq. (7)), the reduction of the case of rational functions to polynomial functions proposed in [16] can be applied:

$$Q_\pi(\mathbb{G},\Omega) = \prod_{x\in\Omega} P(x,\Delta_x^r) - \left(\widetilde{F}_h(\mathbb{G},\Omega)\right)_\pi \prod_{x\in\Omega} P(x,\Delta_x^c)^h \tag{8}$$

Expression $\left(\widetilde{F}_h(\mathbb{G},\Omega)\right)_\pi$ is the constant that results from evaluating $\widetilde{F}_h(\mathbb{G},\Omega)$ at point $\pi$ [16], where $\pi$ is a point of the domain (in our case, $\pi$ represents the probabilities of the rules of 2D-PCFG). Applying the growth transformation method for rational functions to the objective function, stated in expressions (7) and (8), is possible to find an optimum local value. The complete development can be found in Appendix A and in [17], and the final expression is as follows:

$$\bar{p}(A\to\alpha) = \frac{D_{A\to\alpha}(\Delta_x^r) - h\,D_{A\to\alpha}(\Delta_x^c) + p(A\to\alpha)\;\widetilde{C}}{D_A(\Delta_x^r) - h\,D_A(\Delta_x^c) + \widetilde{C}} \tag{9}$$

Term $\widetilde{C}$ must be a constant sufficiently large [16], and the expressions $D_{A\to\alpha}(\Delta_x)$ and $D_A(\Delta_x)$ are given by

$$D_{A\to\alpha}(\Delta_x) = \sum_{x\in\Omega} \frac{1}{P(x,\Delta_x)} \sum_{t_x\in\Delta_x} N(A\to\alpha,t_x)\,P(x,t_x) \tag{10}$$

$$D_A(\Delta_x) = \sum_{x\in\Omega} \frac{1}{P(x,\Delta_x)} \sum_{t_x\in\Delta_x} N(A,t_x)\,P(x,t_x) \tag{11}$$

Following Gopalakrishnan et al. in [16], the development to obtain the expression that allows us to calculate the optimal value of the constant $\widetilde{C}$ can be found in Appendix A and in [17], being its final expression:

$$\widetilde{C} = \max\left\{\max_{p(A\to\alpha)}\left\{-\frac{\left[D_{A\to\alpha}(\Delta_x^r) - h\,D_{A\to\alpha}(\Delta_x^c)\right]}{p(A\to\alpha)}\right\}_\pi, 0\right\} + \epsilon$$

where $\epsilon$ should be a small positive constant.

### 4.4 Discriminative algorithms based on generalized H-criterion

From transformations (9), (10), and (11), a broad family of discriminative learning algorithms for 2D-PCFGs can be defined. This family of algorithms depends on how the respective sets of correct trees $\Delta_x^r$ and competing trees $\Delta_x^c$ are obtained and the values of the parameter $h$. In any case, it must be satisfied that $\Delta_x^r \subset \Delta_x^c$.

The first issue to address is how to compute the set of competing parse trees, $\Delta_x^c$. In this paper, $\Delta_x^c$ will be the set of $N$-best parse trees calculated from the algorithm proposed in [5] and adapted to the 2D-PCFG in [7]. The second issue to consider is how to compute the set of correct parse trees $\Delta_x^r$. In this paper, $\Delta_x^r$, will be the set of $N$-best parse trees by considering only the parse trees compatible with the ground truth.

Below we will show the experiments carried out with the estimation algorithm derived from the implementation of Eqs. (9), (10), and (11) of discriminative learning of 2D-PCFGs based on the generalized H-criterion. We will also study the effect of the $h$ parameter on the result of the proposed discriminative learning algorithm.

## 5 Experimentation

In this section, we will conduct an empirical assessment to analyze the effectiveness of the discriminative learning of 2D-PCFG and their possible application in PrIx extraction tasks.

### 5.1 Datasets and assessment measures

The experiments conducted in this paper were performed with two datasets of different characteristics. First, the *Im2Latex-100k* dataset [14], built for ME recognition tasks, consists of approximately 100 000 LaTeX formulas collected from published articles aggregated in the KDD Cup datasets.[2] The MEs were extracted with regular expressions, filtering out expressions that did not compile or did not fall within the length of 40–1024 characters.

Second, the *IBEM* dataset [15] features ground truth (GT) at different levels, allowing for various types of experiments, such as ME detection and extraction, ME recognition, and ME retrieval. Due to how the *IBEM* dataset has been built by extracting the GT from complete documents of the KDD Cup collection, the MEs compiled in this corpus feature no length restrictions. The ground truth distinguishes between ME embedded into the text (referred to as *in-line*) and

[2] KDD Cup: https://kdd.org/kdd-cup.

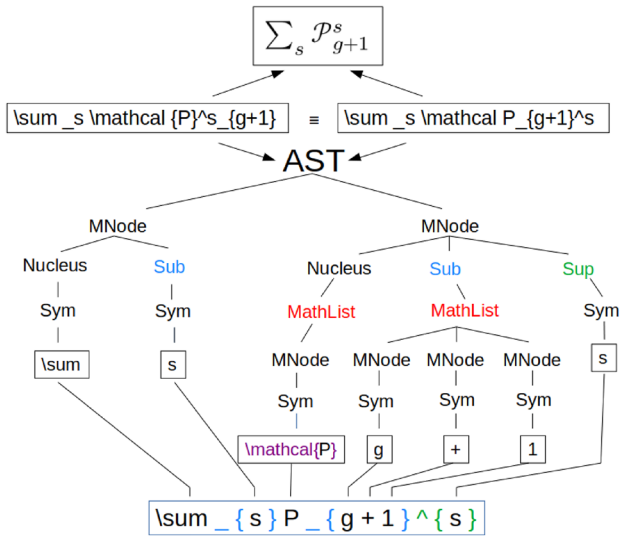


**Fig. 2** Simplification of an AST obtained by parsing a LaTeX ME. The inputs of the normalization process are two equivalent LaTeX markup expressions that render the same ME. The normalized LaTeX markup language is generated by traversing the AST in a depth-first root-left-right-root order. In this example, three normalization steps are undertaken: font normalization (node colored in magenta), sub- and superscript fixed order (nodes colored blue and green), and flattening unnecessary elements of the ME structure (nodes colored in red)

isolated ME (referred to as *displayed*). The dataset consists of approximately 137 000 in-line MEs and 29 000 displayed MEs, with a total of more than 166 000 LaTeX formulas, extracted from over 8 200 pages contained in 600 STEM documents.

In LaTeX, the same ME can be written in several ways. To reduce this structural ambiguity and thus reduce the complexity of the ME recognition task, we have developed two normalizations and filtering processes carried out on both the *Im2Latex-100k* and *IBEM* collections. In the first filtering process (STEP-1 in Table 1), we implemented a ME LaTeX parser that converts the ME LaTeX markup into an abstract syntax tree (AST) for normalization purposes. An example of this filtering process is shown in Fig. 2. In this step, ME that featured complex elements such as arrays, matrices, tables, or images was filtered out.

The ME parser is based on the LuaTeX package *nodetree*,[3] that traverses and visualizes the structure of ME as parsed by the LuaTeX engine.[4] In this AST representation, the inner nodes represent the structural relationships of the ME, and the leaves represent the glyphs to be rendered, which are mapped to LaTeX commands/symbols. It is essential to mention that when parsing ME, all user-defined

[3] Nodetree: https://ctan.org/pkg/nodetree?lang=en.

[4] LuaTeX: http://www.luatex.org/documentation.html.

**Table 1** The table represents the number of samples in each partition, separated by dataset

| Dataset | Filter | Train | Validation | Test |
|---|---|---|---|---|
| | Original partitions [14] | 83 883 | 9 319 | 10 354 |
| *Im2Latex-100k* | STEP-1 | 71 099 | 7 908 | **9 596** |
| | STEP-2 | **50 110** | **5 515** | – |
| | Original partitions | 103 938 | 18 370 | 44 384 |
| *IBEM* | STEP-1 | 100 667 | 17 776 | **42 649** |
| | STEP-2 | **99 125** | **17 548** | – |

Numbers in boldface show the amount of data used in this paper

macros used to abbreviate the formal notation and simplify typesetting are entirely expanded. This expansion of macros is significant for the ME compiled in the *IBEM* dataset, given that user-defined macros were frequent as entire documents were processed.

Besides font normalization, sub- and superscript order fixing, and flattening of the optional grouping of symbols, shown in Fig. 2, the different user-defined horizontal spacing commands were mapped to the `\hspace` command. These normalization steps aim to reduce the inconsistencies and noise in the ME LaTeX markup.

The second filtering process (STEP-2 in Table 1) is related to the fact that our original 2D-PCFG cannot process all the expressions in the *Im2Latex-100k* and *IBEM* datasets. The expressions that could not be parsed were filtered out from the training and validation sets. However, this second filter was not applied to the test set. Table 1 shows the number of samples of the training, validation, and test sets for the original partitions proposed by the authors of the datasets and after the filtering processes STEP-1 and STEP-2. Onward, all experiments performed in this paper have been carried out with the suggested normalization and filtering process.

The recognition process considered in this paper is restricted by 2D-PCFG, which takes as input a whole ME. It is important to note that in other recognition problems like automatic speech recognition, the recognition process is restricted by local information conveyed by n-grams (or, equivalently, by finite-state automata). Since 2D-PCFG takes as input a whole ME, it makes sense to provide information about the size distribution of the ME. Providing correct solutions to small (usually in-line) ME is more accessible than to large ME.

Next, we analyze the average length of the MEs in the training set. Figure 3 shows the histogram indicating the number of MEs in the different length (number of connected components or symbols) intervals. As can be seen in this histogram, the two datasets have very different characteristics. The MEs compiled in the *Im2Latex-100k* dataset are large *displayed* expressions extracted to train and evaluate ME

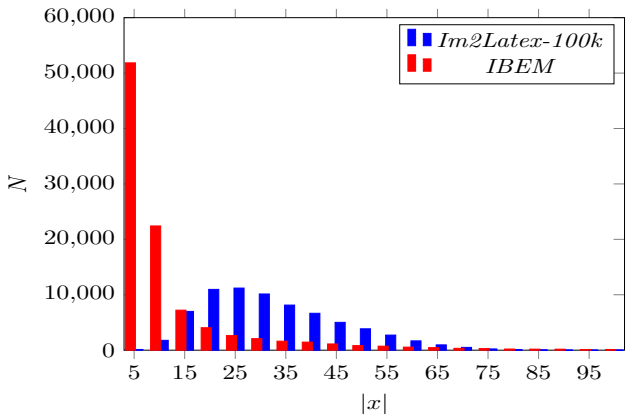


**Fig. 3** Histogram representing the frequency of MEs ($N$) in training set as a function of the different sizes of the MEs ($|x|$). The MEs have been grouped into consecutive intervals of length 5

recognition systems. In contrast, the *IBEM* dataset features many *in-line* shorter expressions, besides a similar distribution for large *displayed* expressions. This characteristic of the *IBEM* dataset introduces a real-world scenario for ME retrieval tasks, in which complete documents are processed, and all MEs are considered, either *in-line* or *displayed*. Figure 3 shows that there are very few expressions with more than 70 symbols in the training set. This same behavior was also observed in the validation set.

For the evaluation of the impact of the estimation algorithms on 2D-PCFG and the viability of our proposal, we consider three different metrics:

- **Exact Accuracy** (*ExAcc*): This metric measures the number of MEs in which the generated hypothesis (1-best or $N$-best) is an exact match with the ground truth (LaTeX expression) of the dataset. Note that this metric is very pessimistic given the average size of the MEs (see Fig. 3) and that we evaluated at the character level. For example, the edit distance between reference "`a_{i}^{2}`" and hypothesis "`a_{i}^{2}`" is two deletions.
- **Bleu**: *Bleu* score [38] measures the difference between the best model prediction and the reference in terms of n-gram precision. This metric measures how far the predictions are from the reference. It is important to remark that we performed tokenization on the ME. This tokenization means that a ME like $a_i^2$ has the GT as "`a_{i}^{2}`" and not as "`a_{i}^{2}`." This measure may be very relevant in PrIx because it measures the precision at the sub-expression level.[5]
- **Levenshtein Distance** (*LevD*): This distance measures the number of insertions, deletions, and substitutions required to match a hypothesis to the reference.

The *LevD* can be confidently computed because of the normalization process that converts any ME in its associated AST. *Bleu* and *LevD* are used in this paper because, in the PrIx context, it is very relevant to evaluate at the sub-expression level, as we mentioned in Sect. 1.

### 5.2 Estimation of 2D-PCFG and parameter setting

We started from the first model (2D-PCFG) obtained from the SESHAT system [13].[6] Considering the particular characteristics of the considered datasets, *Im2Latex-100k* and *IBEM*, we extended this first model to include the necessary rules to account for all symbols and relations appearing in the training sets (see Sect. 5.1). We defined the probabilities of these new rules as equiprobable. The resulting model was our initial baseline model ($\mathbb{G}_i$).

Next, we estimated $\mathbb{G}_i$ using the discriminative learning algorithm based on the generalized H-criterion. As discussed in Sect. 4, this estimation algorithm implements Eqs. (9), (10), and (11). In order to carry out this estimation process, it is necessary first to describe how to calculate the set of correct parse trees $\Delta_x^r$, and the set of competing parse trees $\Delta_x^c$. To obtain $\Delta_x^c$,, we used a new version of the $N$-best parsing algorithm of 2D-PCFG described in [7]. The experiments reported in this research were carried out for $N = 50$. To get $\Delta_x^r$,, we developed a forced version of the parsing algorithm that uses the GT expression and the ME image to generate the most likely reference parse tree using the estimated 2D-PCFG. The estimated model was our final model ($\mathbb{G}_d$). We now discuss several aspects of the estimation algorithm.

The first experiment analyzes the *quality* of $N$-best parse trees, for different values of $N$, on the estimated 2D-PCFG $\mathbb{G}_d$. To do this, we calculated the *Bleu* and *ExAcc* scores in each iteration of the estimation algorithm for a subset of 10 000 randomly selected ME of the *Im2Latex* dataset. This subset was obtained to execute multiple small experiments in a faster way. Figure 4 shows the results of *Bleu* and *ExAcc* in the validation set for $N = \{1, 5, 10, 25, 50\}$ generated by the estimated model ($\mathbb{G}_d$). The *Bleu* and *ExAcc* scores reported in the figure are the minimum among the reference and one of the $N$-best solutions. We report the results for $N$-best hypotheses because this is relevant in the PrIx context: The 1-best solution could not be the correct solution, but it could

[5] It is usual in MT to compute the *Bleu* up to 4 grams when dealing with words. We consider an open problem to be researched in the future to decide the appropriate $n$ of the *Bleu* score for ME evaluation because of the misconception of "word" in ME recognition.

[6] https://github.com/falvaro/seshat.

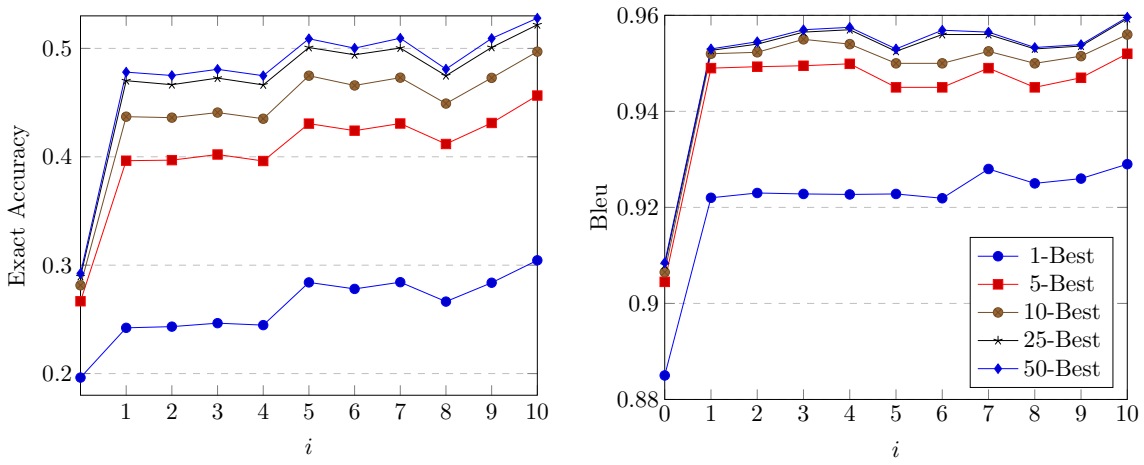


**Fig. 4** *Exact accuracy* and *Bleu* score considering a set of *N*-best hypotheses ({1, 5, 10, 25, and 50}-best) in each iteration *i* in the estimation process. Figure from [19]

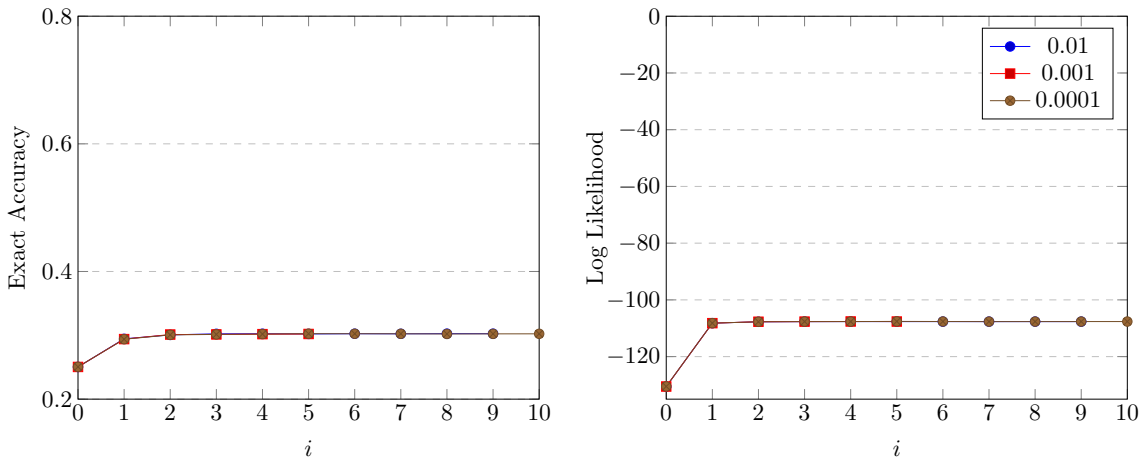


**Fig. 5** Impact of different values of $h$ (0.01, 0.001, 0.0001) in the convergence process with $\epsilon = 0.00001$. Evolution of the *exact accuracy* score and *log likelihood* in the validation set during 10 iterations

be included in one of the following *N*-best hypotheses. As can be seen, the general results improved significantly from the 5-best hypotheses. These results show that considering at least the 5-best hypotheses would be good enough for further use in PrIx. It is also important to remark that only few iterations were needed to get good results, which is one advantage of using discriminative estimation techniques. We observe a parallel behavior for large values of *N*. We suspect that this happens because the *N*-best solutions include many similar solutions with small changes in the leaves of parse trees.

The second experiment aims to optimize the parameters that regulate the discriminative estimation process, $h$ and $\epsilon$ (see Sect. 4). For this, we use the previously selected subset of the *Im2Latex* dataset to be able to analyze the effect of different values for $h$ and $\epsilon$. From the initial grammar ($\mathbb{G}_i$), we obtain different discriminatively estimated grammars for different values of $h$ (0.01, 0.001, 0.0001) and $\epsilon$ (0.0001, 0.00001, 0.000001). As shown in Fig. 5, the convergence of the training process does not change significantly for either exact accuracy or log likelihood. Similar results were obtained by varying the parameter $\epsilon$. Furthermore, it can also be seen that the algorithm converges in less than 5 steps. In our opinion, this effect is due to the small size of the considered 2D-PCFGs (around 450 rules).

The purpose of the third experiment was to estimate the 2D-PCFG parameters. To do this, we selected the values of $h = 0.01$ and $\epsilon = 0.00001$ and considered all training data of both datasets (*Im2Latex* and *IBEM*). For comparison purposes, we also implemented a Viterbi-based estimation

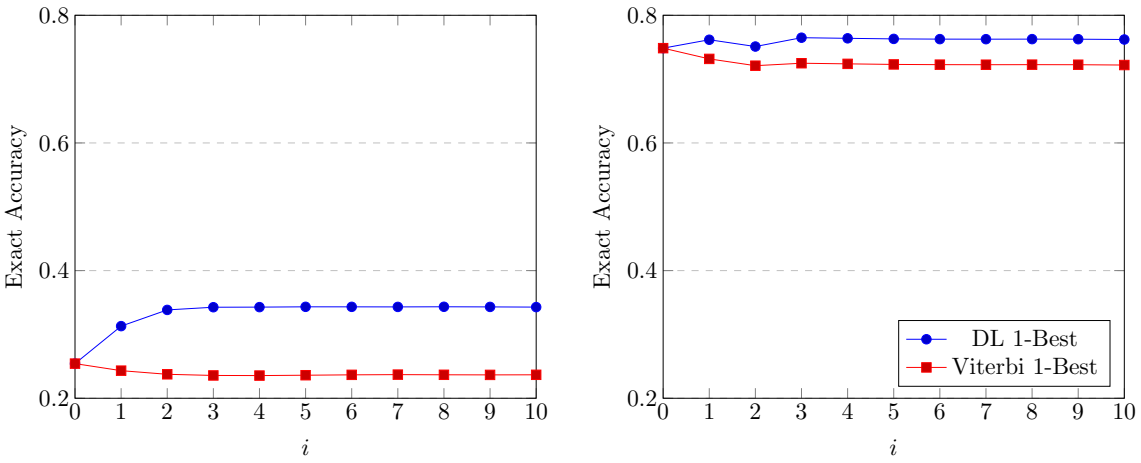


**Fig. 6** 2D-PCFG estimation process by discriminative and Viterbi-based learning on *Im2Latex* (Left) and *IBEM* (Right) datasets. Evolution of the *exact accuracy* score in the validation set during 10 iterations

algorithm. In both cases, we ran 10 iterations for each of the algorithms. Figure 6 shows the evolution of the exact accuracy score in the validation set during the convergence process for the discriminative estimation algorithm and the Viterbi-based estimation algorithm, both for the *Im2Latex* and the I *IBEM* datasets. As can be seen, the discriminative estimation approach provides better performance since it uses much more information from the training samples. This result is consistent for both datasets. It is important to note the difference in the *ExAcc* score in the two datasets. This difference is due to the size distribution on the ME: *IBEM* contains a large amount of small ME that are correctly recognized. This issue is analyzed in the following section.

Finally, we present comparative experiments with other authors on the same corpus. Table 2 shows the results reported by other approaches on the *Im2Latex* corpus together with those obtained by our estimated models with 1-best and 5-best for both the Im2Latex corpus and the IBEM corpus. As can be seen, the results of our approach are not competitive against the state-of-the-art approaches on the *Im2Latex* corpus.[7] These state-of-the-art approaches are based on the end-to-end neural networks technique, which obtains the LaTeX transcript directly from the input image. In our approach, we obtain the LaTeX transcription and generate the syntactic structure associated with said transcription of the ME. This structure is helpful in retrieval problems for searching math subexpressions and for semantic comparisons of formulas where two different MEs could represent the same structure but with different variable names.

**Table 2** The table represents the main experimental results on de *Im2Latex* and IBEM datasets. It shows the Bleu score normalized respect to the size of the formulas (Bleu), the exact accuracy score after deleting whitespace (Match), and the normalized edit distance (*LevD*)

| Dataset | Model | Bleu | Match | LevD |
|---|---|---|---|---|
| | INFTY [39] | 66.65 | 26.66 | – |
| | CTC [3] | 30.36 | 9.16 | – |
| | CAPTION [40] | 75.01 | 55.72 | – |
| *Im2Latex-100k* | IM2TEX-TOK [14] | 73.97 | 77.04 | – |
| | IM2TEX [14] | 87.73 | 79.88 | – |
| | I2L-STRIPS [41] | 88.19 | 68.03 | 0.0725 |
| | OURS 1-Best | 81.26 | 34.62 | 0.1336 |
| | OURS 50-Best | 83.80 | 40.54 | 0.1140 |
| *IBEM* | OURS 1-Best | 71.23 | 53.06 | 0.1651 |
| | OURS 50-Best | 77.16 | 60.14 | 0.1348 |

## 5.3 Experiments depend on the length of MEs

To analyze the ME size's effect on our models' performance, we calculated the *ExAcc* score for different expression lengths in the *Im2Latex* and *IBEM* datasets. Figure 7 shows the obtained results. In all cases, we compared the performance of the discriminatively estimated model ($\mathbb{G}_d$) with the initial model ($\mathbb{G}_i$). Furthermore, we also analyzed the results considering the 1-best and the 50-best hypotheses. As can be seen, the results of the estimated models ($\mathbb{G}_d$) are consistently better than those of the initial model ($\mathbb{G}_i$). As might also be expected, the results considering the

[7] Note that this could be conditioned by our method to tokenize the GT of the ME.

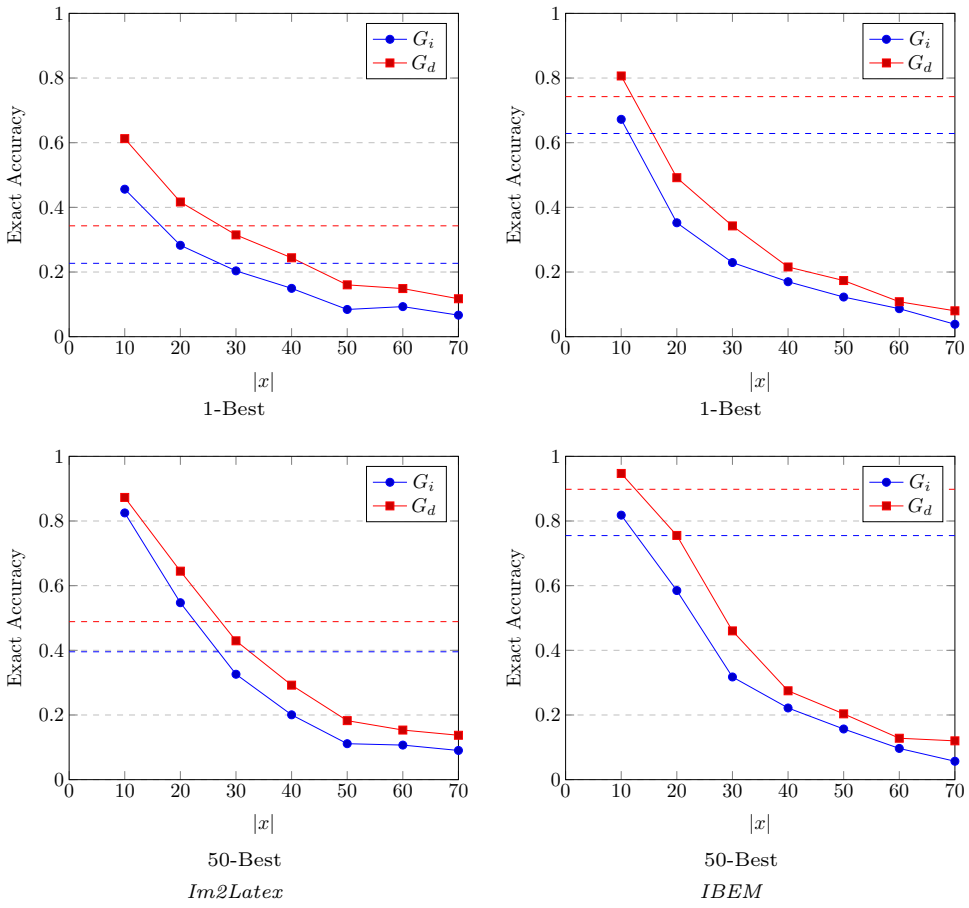


**Fig. 7** Results of *ExAcc* score for the $\mathbb{G}_i$ and $\mathbb{G}_d$ models, considering MEs of different lengths in the *Im2Latex* dataset (Left) and the *IBEM* dataset (Right) using only the 1-best hypothesis (Top) and the 50-best (Bottom). The dotted lines represent the global accuracy of each model for all lengths

50-best hypotheses consistently improve those obtained with the 1-best hypothesis. Similar results were obtained with the *Bleu* score.

Figure 7 shows a generalized drastic decrease in the performance of our models as the length of the expressions increases. This result is unsatisfactory for ME recognition systems. However, in problems of information retrieval or indexing and searching for ME, these results do not seem all that discouraging. In indexing and search problems, we can reasonably assume that the queries will be short expressions or parts of a longer expression.

## 5.4 Error-dependent precision

As mentioned above, *ExAcc* is a very harsh metric requiring an exact comparison symbol by symbol and at the same positions. Note that this measure is also very dependent on the way of preparing the GT. Since we aim to address search problems, we could consider some relaxation on this measure. For this purpose, we explored the Levenshtein distance as a measure that allows us to analyze the number of admissible errors. Figure 8 shows the *precision* of the discriminatively estimated model ($\mathbb{G}_d$) and the initial model ($\mathbb{G}_i$), varying the number of admissible errors for the *Im2Latex* and *IBEM* datasets. As in previous cases, these results have been obtained with the 1-best and the 50-best hypotheses. The plots show how most results, even for very long expressions, are very close to the reference when we allow a certain number of admissible errors. These results reinforce the possible practical use of admissible errors to optimize a search engine's precision–recall. To illustrate this reasoning, Fig. 9 provides one example of a long expression where the model hypothesis is incorrect but only by one relationship error. In classic search problems, most of the queries (subexpressions of this expression) could still find the reference.

To conclude, we present the final experiments on the test set of the *Im2Latex* dataset. We selected this corpus as our models reported worse performance with it (see Fig. 6 and 7). Figure 10 shows the *precision* with the estimated model ($\mathbb{G}_d$) varying the number of admissible errors. As can be seen, the results are reasonable, although somewhat worse than those reported on the validation set. However, it should be noted that the STEP-2 filter is not applied to the test set (see Table 1).

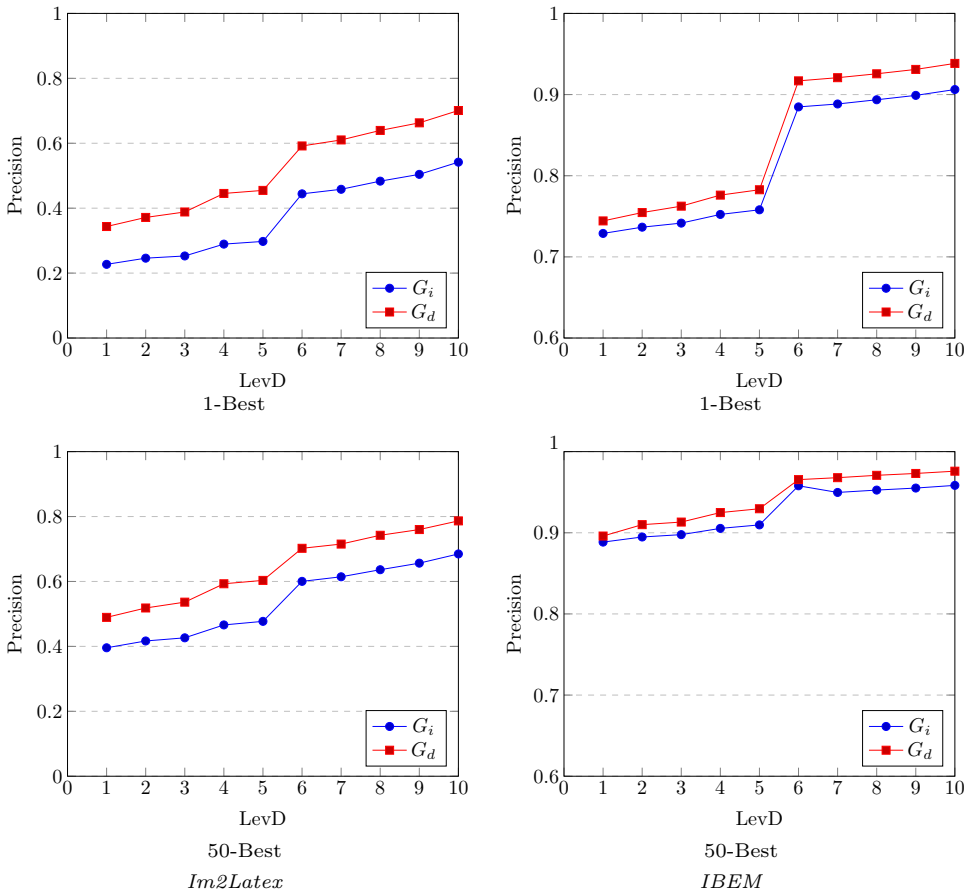


**Fig. 8** *Precision* results for the $\mathbb{G}_i$ and $\mathbb{G}_d$ models, considering the different maximum number of admissible errors in the *Im2Latex* dataset (Left) and the *IBEM* dataset (Right) using only the 1-best hypothesis (Top) and the 50-best (Bottom)

$$\left[\frac{1}{2}x + (3-x)^2 + (3+x)^2\right]^{\frac{1}{2}} = \left(\frac{1}{x} - \frac{1}{2}x^2 + 5\right)^2,$$

(a) Reference

$$\left[\frac{1}{2}x + (3-x)^2 + (3+x)^2\right] \frac{1}{2} = \left(\frac{1}{x} - \frac{1}{2}x^2 + 5\right)^2,$$

(b) Hypothesis

**Fig. 9** The figure shows an example of an incorrect prediction. This prediction has a Levenshtein distance equal to 1, where the model mistakes the subexpression $^{\frac{1}{2}}$ with $\frac{1}{2}$

## 6 Conclusions

This paper presents a discriminative learning algorithm to estimate a 2D-PCFG based on a generalization of the H-criterion as the objective function and the growth transformations as the optimization method. Several experiments have been reported on two well-known datasets. In the experiments that analyze the convergence of the estimation algorithm, the results improve significantly from the 5-best hypotheses. These results would be enough to compose a hypergraph for further use in PrIx.

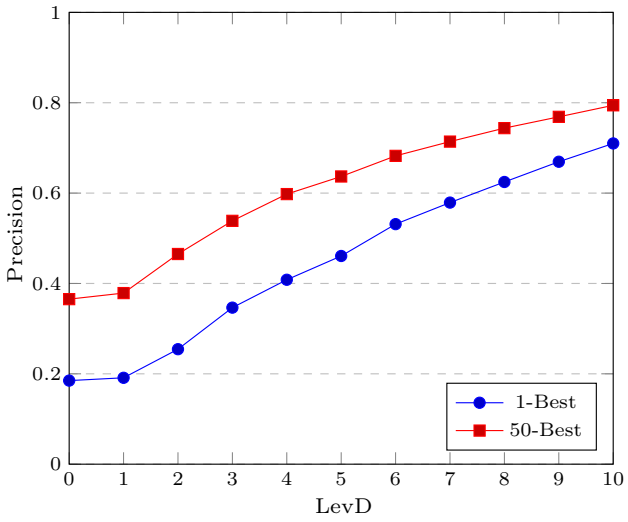


**Fig. 10** *Precision* results of the $\mathbb{G}_d$ model on the test set of the Im2Latex dataset for different error values in the *LevD* using the 1-best and 50-best on the test set. Figure from [19]

In the experiments related to the length of the MEs, the model precision drops significantly beyond a ME length of |30|. However, using a structural model allows us to easily generate multiple hypotheses and decompose large expressions into all their correct subexpressions. Thanks to this and given that queries in PrIx may be of short length (less than 20 LaTeX symbols), the mistakes produced by the model could not affect the majority of queries. Considering all these, our model provides a good approximation to the PrIx problem in massive collections of digitized scientific documents. We expect to research this PrIx framework in future work.

Extending the results of our previous work [19], we introduced a comparison of the discriminative estimation algorithm with Viterbi estimation in the *Im2Latex* corpus and a new *IBEM* corpus with different characteristics. This experiment shows that our proposal obtains better results for the task of mathematical expression recognition than the Viterbi baseline. Further experiments in the model training also showed that the small grammar converges in a few iterations and is not significantly affected by changes in $h$ and $\epsilon$. After optimizing hyperparameters, the final results are displayed beside previous models, showing that the model obtains similar results to other models that are not end-to-end neural networks.

## A Appendix: optimization by growth transformations for generalized H-criterion

In this appendix, we will develop in detail the estimation process of the model parameters (rules probabilities, $p(A \to \alpha) \in \mathcal{P}$ of $\mathbb{G}$, see Def. (3)) until reaching expression (9). To begin with, we present the growth transformation optimization framework and how to optimize Eq. (8) by applying the growth transformations for rational functions [16]. In each iteration of the optimization process, the update parameters $p(A \to \alpha)$ are obtained using the following expression:

$$\bar{p}(A \to \alpha) = \frac{p(A \to \alpha)\left[\frac{\partial Q_\pi(\mathbb{G},\Omega)}{\partial p(A \to \alpha)} + \mathrm{C}\right]_\pi}{\sum_{i=1}^{n_A} p(A \to \alpha_i)\left[\frac{\partial Q_\pi(\mathbb{G},\Omega)}{\partial p(A \to \alpha_i)} + \mathrm{C}\right]_\pi} \tag{12}$$

$n_A$ is the number of rules with the non-terminal $A$ as the left side of the rule, and $\pi = (\pi_{A_1}, \pi_{A_2}, \ldots, \pi_{A_{|N|}})$: $A_i \in N$, $1 \le i \le |N|$ is a vector defined as follows: $\pi_{A_i} = (p(A_i \to \alpha_{i1}), p(A_i \to \alpha_{i2}), \ldots, p(A_i \to \alpha_{in_{A_i}}))$. Furthermore, $Q_\pi(\mathbb{G}, \Omega)$ (see Eq. (8)) is a polynomial function. As it was demonstrated in [16], for every point of the domain $\pi$, there is a constant C such that the polynomial $P_\pi + \mathrm{C}$ has only non-negative coefficients. Following a similar development to that used in [11], we will allow us to obtain $\bar{p}(A \to \alpha)$ from Eq. (12).

First of all, let us define an auxiliary function

$$\mathcal{D}^h_{A\to\alpha}(\Delta_x) = p(A \to \alpha)\left[\frac{\partial \prod_{x\in\Omega} P(x,\Delta_x)^h}{\partial p(A \to \alpha)}\right]_\pi$$

then expression (12) can be rewritten as

$$\bar{p}(A \to \alpha) = \frac{\mathcal{D}^1_{A\to\alpha}(\Delta^r_x) - \left(\widetilde{F}_h(\mathbb{G},\Omega)\right)_\pi \mathcal{D}^h_{A\to\alpha}(\Delta^c_x) + p(A \to \alpha)\ \mathrm{C}}{\sum_{i=1}^{n_A} \mathcal{D}^1_{A\to\alpha_i}(\Delta^r_x) - \left(\widetilde{F}_h(\mathbb{G},\Omega)\right)_\pi \sum_{i=1}^{n_A} \mathcal{D}^h_{A\to\alpha_i}(\Delta^c_x) + p(A \to \alpha_i)\,\mathrm{C}} \tag{13}$$

We will begin by developing the expression $\mathcal{D}^h_{A\to\alpha}(\Delta_x)$ as a preliminary step to evaluating the expressions $\mathcal{D}^1_{A\to\alpha}(\Delta^r_x)$ and $\mathcal{D}^h_{A\to\alpha}(\Delta^c_x)$ of the numerator.

$$\begin{aligned}
\mathcal{D}^h_{A\to\alpha}(\Delta_x) &= \\
&= p(A \to \alpha)\left[h \prod_{x\in\Omega} P(x,\Delta_x)^{h-1} \frac{\partial \prod_{x\in\Omega} P(x,\Delta_x)}{\partial p(A \to \alpha)}\right]_\pi \\
&= h\left[\prod_{x\in\Omega} P(x,\Delta_x)^h \sum_{x\in\Omega} \frac{p(A \to \alpha)}{P(x,\Delta_x)} \frac{\partial P(x,\Delta_x)}{\partial p(A \to \alpha)}\right]_\pi \\
&= h\left[\prod_{x\in\Omega} P(x,\Delta_x)^h \sum_{x\in\Omega} \frac{1}{P(x,\Delta_x)} \sum_{t_x\in\Delta_x} N(A \to \alpha, t_x) P(x,t_x)\right]_\pi
\end{aligned} \tag{14}$$

Similarly, we will develop the expression $\sum_{i=1}^{n_A} \mathcal{D}^h_{A\to\alpha_i}(\Delta_x)$ as a preliminary step to evaluating the expressions $\sum_{i=1}^{n_A} \mathcal{D}^1_{A\to\alpha_i}(\Delta^r_x)$ and $\sum_{i=1}^{n_A} \mathcal{D}^h_{A\to\alpha_i}(\Delta^c_x)$ of the denominator.

$$\begin{aligned}
&\sum_{i=1}^{n_A} \mathcal{D}^h_{A\to\alpha_i}(\Delta_x) = \\
&= \sum_{i=1}^{n_A} h\left[\prod_{x\in\Omega} P(x,\Delta_x)^h \sum_{x\in\Omega} \frac{1}{P(x,\Delta_x)} \sum_{t_x\in\Delta_x} N(A \to \alpha_i, t_x) P(x,t_x)\right]_\pi \\
&= h\left[\prod_{x\in\Omega} P(x,\Delta_x)^h \sum_{x\in\Omega} \frac{1}{P(x,\Delta_x)} \sum_{t_x\in\Delta_x} \sum_{i=1}^{n_A} N(A \to \alpha_i, t_x) P(x,t_x)\right]_\pi \\
&= h\left[\prod_{x\in\Omega} P(x,\Delta_x)^h \sum_{x\in\Omega} \frac{1}{P(x,\Delta_x)} \sum_{t_x\in\Delta_x} N(A, t_x) P(x,t_x)\right]_\pi
\end{aligned} \tag{15}$$

Given the expression $\widetilde{F}_h(\mathbb{G},\Omega)$ in Eq. (7) and substituting expressions (14) and (15) in transformation (13), we get the final expression after simplifying $\prod_{x\in\Omega} P(x,\Delta^r_x)$ in the numerator and denominator.

$$\bar{p}(A \to \alpha) = \frac{D_{A\to\alpha}(\Delta^r_x) - h\, D_{A\to\alpha}(\Delta^c_x) + p(A \to \alpha)\frac{\mathrm{C}}{\prod_{x\in\Omega} P(x,\Delta^r_x)}}{D_A(\Delta^r_x) - h\, D_A(\Delta^c_x) + \frac{\mathrm{C}}{\prod_{x\in\Omega} P(x,\Delta^r_x)}} \tag{16}$$

The auxiliary functions $D_{A\to\alpha}(\Delta_x)$ and $D_A(\Delta_x)$ will be given by

$$D_{A\to\alpha}(\Delta_x) = \sum_{x\in\Omega} \frac{1}{P(x,\Delta_x)} \sum_{t_x\in\Delta_x} N(A\to\alpha, t_x)\, P(x,t_x)\,,$$
$$D_A(\Delta_x) = \sum_{x\in\Omega} \frac{1}{P(x,\Delta_x)} \sum_{t_x\in\Delta_x} N(A, t_x)\, P(x,t_x)\,. \tag{17}$$

Gopalakrishnan et al. in [16] suggest that to obtain a fast convergence and to guarantee the conditions of the growth transformations theorem for rational functions, the constant C should be calculated as follows:

$$\mathrm{C} = \max\left\{ \max_{p(A\to\alpha)} \left\{ -\frac{\partial Q_\pi(\mathbb{G},\Omega)}{\partial p(A\to\alpha)} \right\}_\pi, 0 \right\} + \epsilon \tag{18}$$

where $\epsilon$ is a small positive constant. Considering the expression $Q_\pi(\mathbb{G},\Omega)$ in Eq. (8) and carrying out a development similar to the one we have done to obtain Eq. (16), expression (18) is as follows:

$$\frac{\partial Q_\pi(\mathbb{G},\Omega)}{\partial p(A\to\alpha)} = \frac{\prod_{x\in\Omega} P(x,\Delta_x^r)}{p(A\to\alpha)} \left[ D_{A\to\alpha}(\Delta_x^r) - h\, D_{A\to\alpha}(\Delta_x^c) \right]$$

Substituting this expression in Eq. (18) allows us to calculate a C maximum, ($\widetilde{\mathrm{C}}$), as:

$$\widetilde{\mathrm{C}} = \max\left\{ \max_{p(A\to\alpha)} \left\{ -\frac{\left[ D_{A\to\alpha}(\Delta_x^r) - h\, D_{A\to\alpha}(\Delta_x^c) \right]}{p(A\to\alpha)} \right\}_\pi, 0 \right\} + \epsilon$$

Finally, expression (16) becomes

$$\bar{p}(A\to\alpha) = \frac{D_{A\to\alpha}(\Delta_x^r) - h\, D_{A\to\alpha}(\Delta_x^c) + p(A\to\alpha)\ \widetilde{\mathrm{C}}}{D_A(\Delta_x^r) - h\, D_A(\Delta_x^c) + \widetilde{\mathrm{C}}}\,.$$

This expression and auxiliary expressions (17) coincide with expressions (9), (10), and (11) as we had proposed.

**Acknowledgements** This research has been developed with the support of Grant PID2020-116813RBI00a funded by MCIN/AEI/10.13039/501100011033 and FPI grant CIACIF/2021/313 funded by Generalitat Valenciana. Universitat Politècnica de València Grant No. SP20210263

**Funding** Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

**Data availability** The data that support the findings of this study are freely available for research purposes as described in [14] and [15].

## Declarations

**Conflict of interest** All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript.



## References

1. Bahl LR, Jelinek F, Mercer RL (1983) A maximum likelihood approach to continuous speech recognition. IEEE Trans Pattern Anal Machine Intell 5(2):179–190
2. Koehn P (2009) Statistical Machine Translation. Cambridge University Press, ???. https://doi.org/10.1017/CBO9780511815829
3. Graves A, Fernández S, Gomez F, Schmidhuber J (2006) Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In: ICML, vol 2006, pp 369–376. https://doi.org/10.1145/1143844.1143891
4. Marzal A (1993) Cálculo de las k mejores soluciones a problemas de programación dinámica. PhD thesis, Universidad Politécnica de Valencia
5. Jiménez VM, Marzal A (2000) Computation of the N Best Parse Trees for Weighted and Stochastic Context-Free Grammars. In: Advances in Pattern Recognition. Lecture Notes in Computer Science, 1876, pp 183–192 https://doi.org/10.1007/3-540-44522-6_19
6. Ortmanns S, Ney H, Aubert X (1997) A word graph algorithm for large vocabulary continuous speech recognition. Comput Speech Lang 11(1):43–72. https://doi.org/10.1006/csla.1996.0022
7. Noya E, Sánchez JA, Benedí JM (2021) Generation of Hypergraphs from the N-Best Parsing of 2D-Probabilistic Context-Free Grammars for Mathematical Expression Recognition. In: ICPR, pp 5696–5703. https://doi.org/10.1109/ICPR48806.2021.9412273
8. Ueffing N, Och FJ, Ney H (2002) Generation of word graphs in statistical machine translation. In: Proceedings of the 2002 conference on empirical methods in natural language processing (EMNLP 2002), pp 156–163. Association for Computational Linguistics, ???. https://doi.org/10.3115/1118693.1118714. https://aclanthology.org/W02-1021
9. Toselli AH, Vidal E, Puigcerver J, Noya-García E (2019) Probabilistic multi-word spotting in handwritten text images. Pattern Anal Appl 22:23–32. https://doi.org/10.1007/s10044-018-0742-z
10. Sánchez-Sáez R, Sánchez JA, Benedí JM (2010) Confidence measures for error discrimination in an interactive predictive parsing framework. In: Coling, pp 1220–1228
11. Benedí JM, Sánchez JA (2005) Estimation of stochastic context-free grammars and their use as language models. Comput Speech Lang 19(3):249–274. https://doi.org/10.1016/j.csl.2004.09.001
12. Awal AM, Mouchère H, Viard-Gaudin C (2012) A global learning approach for an online handwritten mathematical expression recognition system. Pattern Recogn Lett 35:68–77. https://doi.org/10.1016/j.patrec.2012.10.024
13. Álvaro F, Sánchez JA, Benedí JM (2016) An Integrated Grammar-based Approach for Mathematical Expression Recognition.

Pattern Recogn 51:135–147. https://doi.org/10.1016/j.patcog.2015.09.013
14. Deng Y, Kanervisto A, Ling J, Rush AM (2017) Image-to-markup generation with coarse-to-fine attention. In: Proceedings of the ICML-17, pp 980–989
15. Anitei D, Sánchez JA, Fuentes JM, Paredes R, Benedí JM (2021) ICDAR2021 Competition on mathematical formula detection. In: ICDAR, pp 783–795. https://doi.org/10.1007/978-3-030-86337-1_52
16. Gopalakrishnan PS, Kanevsky D, Nadas A, Nahamoo D (1991) An inequality for rational functions with applications to some statistical estimation problems. IEEE Trans Inf Theory 37(1):107–113. https://doi.org/10.1109/18.61108
17. Maca M, Benedí JM, Sánchez JA (2021) Discriminative Learning for Probabilistic Context-Free Grammars based on Generalized H-Criterion. Preprint arXiv:2103.08656arXiv:2103.08656 [cs.CL]
18. Woodland PC, Povey D (2002) Large scale discriminative training of hidden Markov models for speech recognition. Comput Speech Lang 16(1):25–47. https://doi.org/10.1006/csla.2001.0182
19. Noya E, Benedí JM, Sánchez JA, Anitei D (2022) Discriminative learning of two-dimensional probabilistic context-free grammars for mathematical expression recognition and retrieval. In: IbPRIA, pp 333–347. https://doi.org/10.1007/978-3-031-04881-4_27
20. Zanibbi R, Blostein D (2011) Recognition and Retrieval of Mathematical Expressions. IJDAR 15:331–357. https://doi.org/10.1007/s10032-011-0174-4
21. Huang J, Tan J, Bi N (2020) Overview of mathematical expression recognition. In: Pattern recognition and artificial intelligence, pp 41–54. https://doi.org/10.1007/978-3-030-59830-3_4
22. Mahdavi M, Zanibbi R, Mouchere H, Viard-Gaudin C, Garain U (2019) ICDAR 2019 CROHME + TFD: Competition on recognition of handwritten mathematical expressions and typeset formula detection. In: ICDAR, pp 1533–1538. https://doi.org/10.1109/ICDAR.2019.00247
23. Wang DH, Yin F, Wu JW, Yan YP, Huang ZC, Chen GY, Wang Y, Liu CL (2020) ICFHR 2020 Competition on offline recognition and spotting of handwritten mathematical expressions - OffRaSHME. In: ICFHR, pp. 211–215. https://doi.org/10.1109/ICFHR2020.2020.00047
24. Wan Z, Fan K, Wang Q, Zhang S (2019) Recognition of printed mathematical formula symbols based on convolutional neural network. DEStech Transactions on Computer Science and Engineering. https://doi.org/10.12783/dtcse/ica2019/30711
25. Wu J-W, Yin F, Zhang Y-M, Zhang X-Y, Liu C-L (2020) Handwritten mathematical expression recognition via paired adversarial learning. Int J Comput Vis 128:2386–401. https://doi.org/10.1007/s11263-020-01291-5
26. Peng S, Gao L, Yuan K, Tang Z (2021) Image to LaTeX with Graph Neural Network for Mathematical Formula Recognition. In: ICDAR, pp 648–663. https://doi.org/10.1007/978-3-030-86331-9_42
27. Zhao W, Gao L, Yan Z, Peng S, Du L, Zhang Z (2021) Handwritten mathematical expression recognition with bidirectionally trained transformer. In: Document analysis and recognition – ICDAR 2021, pp 570–584. https://doi.org/10.1007/978-3-030-86331-9_37
28. Davila K, Joshi R, Setlur S, Govindaraju V, Zanibbi R (2019) Tangent-V: Math formula image search using line-of-sight graphs, pp 681–695. https://doi.org/10.1007/978-3-030-15712-8_44
29. Zhong W, Zanibbi R (2019) Structural similarity search for formulas using leaf-root paths in operator subtrees, pp 116–129. https://doi.org/10.1007/978-3-030-15712-8_8
30. Mansouri B, Zanibbi R, Oard D (2019) Characterizing searches for mathematical concepts, pp 57–66. https://doi.org/10.1109/JCDL.2019.00019
31. Chou PA (1989) Recognition of equations using a two-dimensional stochastic context-free grammar. In: Visual communications and image processing IV, vol 1199, pp 852–863. https://doi.org/10.1117/12.970095
32. Průša D, Hlaváč V (2007) Mathematical Formulae Recognition Using 2D Grammars. ICDAR 2, 849–853. https://doi.org/10.1109/ICDAR.2007.4377035
33. Lari K, Young SJ (1991) Applications of stochastic context-free grammars using the inside-outside algorithm. Comput Speech Lang 5(3):237–257. https://doi.org/10.1016/0885-2308(91)90009-F
34. Ney H (1992) Stochastic grammars and pattern recognition. In: Laface, P., De Mori, R. (eds.) Speech recognition and understanding, pp 319–344. https://doi.org/10.1007/978-3-642-76626-8_34
35. Baum LE, Sell GR (1968) Growth transformation for functions on manifolds. Pac J Math 27(2):211–227
36. Casacuberta F (1996) Growth transformations for probabilistic functions of stochastic grammars. IJPRAI 10(3):183–201. https://doi.org/10.1142/S0218001496000153
37. Gopalakrishnan P, Kanevsky D, Nadas A, Nahamoo D, Picheny M (1988) Decoder selection based on cross-entropies. In: ICASSP-88, vol 1, pp 20–23. https://doi.org/10.1109/ICASSP.1988.196499
38. Papineni K, Roukos S, Ward T, Zhu WJ (2002) BLEU: a method for automatic evaluation of machine translation. In: ACL, pp 311–318. https://doi.org/10.3115/1073083.1073135
39. Suzuki M, Tamari F, Fukuda R, Uchida S, Kanahori T (2003) Infty: an integrated ocr system for mathematical documents, pp 95–104. https://doi.org/10.1145/958220.958239
40. Shi B, Bai X, Yao C (2017) An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. TPAMI 39–11:2298–2304. https://doi.org/10.1109/TPAMI.2016.2646371
41. Singh S (2018) Teaching machines to code: neural markup generation with visual attention. Preprint arXiv:1802.05415arXiv:1802.05415 [cs.CL]