A Font Setting Based Bayesian Model to Extract Mathematical Expression in PDF Files

Conference Paper · November 2017

DOI: 10.1109/ICDAR.2017.129

2 authors:

Xing Wang
Texas A&M University
19 PUBLICATIONS **51** CITATIONS
SEE PROFILE

Jyh-Charn Liu
Texas A&M University
31 PUBLICATIONS **142** CITATIONS
SEE PROFILE

On a Font Setting based Bayesian Model to extract mathematical expression in PDF files

Xing Wang
Department of Computer Science and Engineering
Texas A&M University
College Station, USA
xingwang@cse.tamu.edu

Jyh-Charn Liu
Department of Computer Science and Engineering
Texas A&M University
College Station, USA
liu@cse.tamu.edu

***Abstract*— This paper proposes a Font Setting based Bayesian (FSB) model to extract mathematical expressions (MEs) in the portable document format (PDF) files. Without explicit tags for MEs, MEs have highly diverse styles, and multiple font setting techniques can be used to express one particular printed ME in the PDF files. The FSB model is a self-adaptive unsupervised algorithm which first uses rules to identify ME and none-ME (NME) and then extracts the remaining ME using the Bayesian inference based on the observation that MEs tend to repeatedly represented in a particular style. PDF files are first processed using parser and document layout are analyzed using projection profiling cutting based algorithm to detect columns and lines. Next, heuristic rules which are derived from the knowledge of math usage and common practices are employed to reason about the posterior probability of a char as ME and also as Non-ME, conditional on the font and value information. Based on the char level posterior probability, Bayesian inference is applied to infer the none-separable character set (NSCS) being ME or not. Consecutive ME NSCS are merged as ME. Experimental results based on a few heuristic rules, which can be easily expanded, for ME extraction show that our approach achieves higher F1 score than that of the state of art solutions for word level ME classification and lower miss/false rate for ME extraction. The processing time is markedly shorter than supervised machine learning techniques, and the extracted information and analytics products can be readily used for creation of high level applications.**

Keywords-Font; Mathematical expression extraction; PDF document; layout analysis

I. INTRODUCTION

The vast majority of modern publications are made available online in the digital form, most of which are processed into the Portable Document Format (PDF). Being able to extract the mathematical expressions (ME) from PDF files is particularly useful for the Science, Technology, Engineering, and Mathematics (STEM) fields, because they offer the succinct and usually most unambiguous abstractions of complex subjects. Commonly, real world objects are defined as mathematical symbols or variables, the interaction between objects by mathematical operations, and relationship between objects by mathematical relation. Being able to separate MEs from their non-ME in STEM papers offers a novel knowledge abstraction of complex STEM papers [1] [2], and some noted efforts are underway [3] [4] [5] [6].

The PDF file format is designed for the printing systems, where the only distinction between ME vs. NME is that MEs are usually expressed by certain styled fonts e.g., sizes, symbols, spacing, and capitalization to offer distinct visual effects, but no explicit tag is available to differ ME from NME ME can be further divided into Isolated MEs (IME) (or offset expression [2]), which are explicitly separated from the plaintext part, and the Embedded MEs (EME), which are usually treated as a form of technical entity being blended into plaintext sentences for reasoning, explanation, association of the mathematical notions and the subject under discussion, etc. It is relatively easy to extract IME because of their highlighted spacing and style. On the other hand, EME extraction is much more challenging, with the best published performance at the level of false negative rate of 15.9%, and the false positive rate over 20% [6].

Main challenges for extraction of EME extraction include the diverse usage of font setting techniques, including importing of special fonts, and unrestricted use of fonts for MEs and NME. To overcome these challenges, we observe that the authors in a paper tend to express MEs in a particular style repeatedly. When combined with the common practices in ME related technical writing, they can be used as screening rules to match partial NME and ME. To take advantage of these properties, in this paper we propose a rule based system for IME detection and an unsupervised Font Setting based Bayesian (FSB) model for EMEs extraction. A toolchain is developed to implement the FSB model. For page layout analysis, a Projection Profiling Cutting (PPC) based algorithm is designed for Line Column Generation (LCG) based on the image converted from PDF files. Multiple semantic resources that include natural language corpus, citation style, headings, highlighting words, math symbols, and math function names, are ensemble into heuristic rules for detecting anchoring MEs and NMEs, which represent the entities that can be recognized with negligible error. The anchoring ME and NME are used to estimate the conditional probability of a character as ME, which will then be further used to extract ME in the whole document through Bayesian inference.

he FSB model is evaluated using a public dataset, which is manually verified by the first author to create the ground truth for the experiments. The FSB model outperformed existing machine-learning approaches [6] for the F1 score comparison by 7 percentages for the word level classification. The rule based IME extraction module achieves the precision of 99.4% and slightly better F1 score than the best experiment setting of

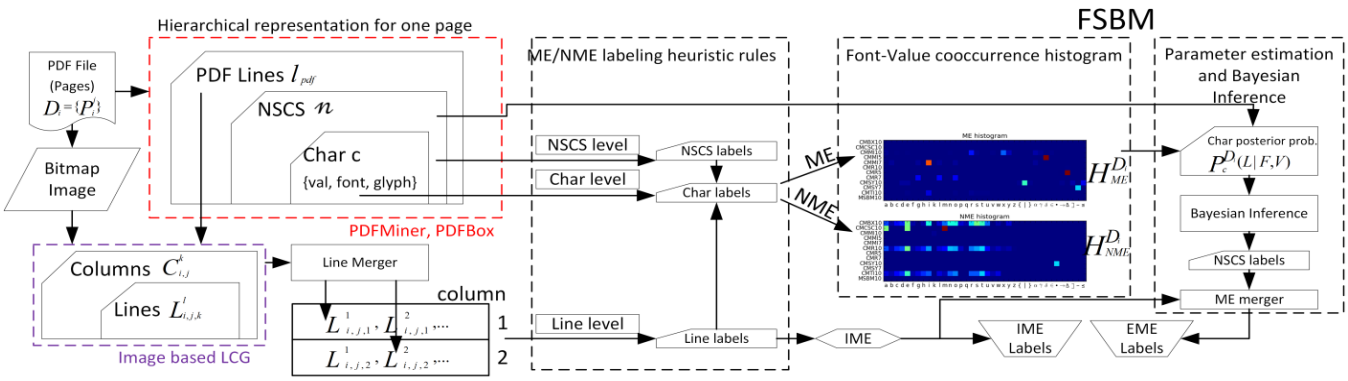

Figure 1. The system architecture of the FSB System. The font value co-occurrence matrix is derived from test file 10.1.1.6.2203_3 [13]

previous work. As for the EME extraction, the FSB model made 5 percent gains over the state of art solution in both the missing rate and the false rate.

The rest of the paper is organized as follows. Section II presents related work. Section III presents document model, the FSB model, and the heuristic rules to identify MEs and NMEs. Section IV explains experiment settings, dataset, and evaluation criteria, shows the performance, and gives a discussion. We summarize our contribution in conclusion in Section V.

II. Related Work

We will present related work from three aspects: document layout analysis, ME identification, and Non-ME identification resources.

Before we extract MEs, it is necessary to process the document into a hierarchical layout representation such as column, line and words. As pointed out by Lin [5], the quality of text line segmentation highly affects the performance of IME detection. Projection profiling cutting (PPC) is a widely adopted technique for document analysis [7] and mathematical analysis [8]. Unlike universal document layout analysis where the skew is an important issue to consider, we can assume that the printed image converted from a PDF file only contains horizontal lines with little noise. Based on the concept of PPC together with some aforementioned heuristics, we proposed a simple formulation for the problem of double-column detection and line segmentation.

Isolated ME and embedded ME Identification have common features and distinct features as clues. Math symbols [3] [9], including binary relations, operations, Greek symbols, delimiters, functions, integrals, fractions, and squares are the most obvious indicators. Named functions are also used as clues [3]. Font is also a common factor for both, including size [10], italics [3], variance of font size [3], and special font-names [11]. That being said, these cues are not sufficient for ME detection and they are incapable of handing custom fonts. Lin [4] also proposed the purity of words to check the alphabet percentage. Besides, IME and EME have their own special features. IME is usually labeled with formula serial number [3]. There are also some special layout features, including: line height, space above line, space below line, left indent right indent [12], line centerness, variation of line width [3], sparseness of characters, variance of baseline, variance of bounding box (space/area, width, height) [4]. At last, in addition to clues about the elements themselves, the neighbors also play an important role. For example, [4] used the label of neighbor as a feature and [9] used the context as semantic constraints and made an assessment of the relation between connected characters [9]. The ICST dataset [13] is a public dataset for evaluation and it also provides a set of evaluation criteria and the evaluation script.

For NME detection, we adapt the grammar for figure id detection [14] and adopt it for the table id, equation id detection. Natural Language Processing resources from the Natural Language ToolKit (NLTK) [15] are used to identify the plain text. Some work attempted to determine on whether a sequence of char as plain words [16] [4], but they only decide based on whether the consecutive chars are only alphabetic.

III. The FSB Model and System

As shown in Fig. 1, the system is composed of three components. A PDF file is first parsed into our document model to be discussed in subsection A. Next, as to be discussed in subsection A.2, open source packages are adopted to extract character information and a rough hierarchical representation: PDF line, Non-Separable Character Set (NSCS), char. Besides, accurate column and lines will be extracted using an image based Line-Column-Generation (LCG) algorithm in subsection A.3. A Line Merger module will merge the result from the above two sub-modules. The second component, elaborated in subsection B, detects ME/NME using heuristic rules. At last, we will present our Font Setting based Bayesian (FSB) Model and the procedures to estimate the parameters in subsection C. The inference of ME vs. NME will be conducted at the NSCS level, and the ME merger will do a post processing to get the final MEs.

A. Document Model and PDF processing

1) Document Model

A PDF document D_i is composed of many pages $\{P_i^j\}$, where each page P_i^j consists of columns $\{C_{i,j}^k\}$. Each column is composed of lines $\{L_{i,j,k}^l\}$, where the line could represent a

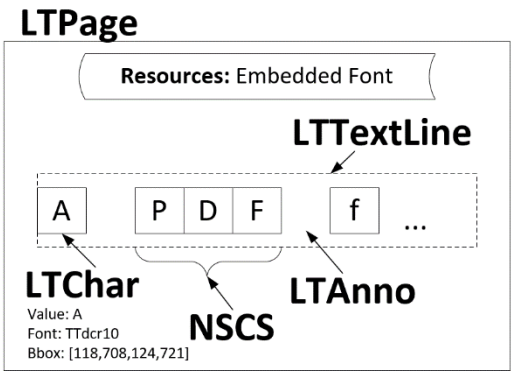

Figure 3. Elements in PDF file from PDFMiner

plain text line, an IME, a figure, or a table, etc. Each line $L^{l}_{i,j,k}$ is composed of characters which could be organized as a sequence of Non-Separable Character Set (NSCS), $(n^{1}_{i,j,k,l}, n^{2}_{i,j,k,l}, \dots)$. A NSCS n could correspond to a plain text word or part of a ME. For each character $c \in n$, it is associated with a text value $v_c \in [0,255]$, a Unicode value u_c, a font $f_c \in \mathcal{F}_i$, and a bounding box b_c, where font set $\mathcal{F}_i$ is one type of resource for document D_i. Each bounding box is represented by a quad-tuple of the "left", "bottom", "right", "top" of a rectangle, together with the origin position in the left-bottom corner of a page.

2) PDF Parsing

A PDF file is parsed using PDFMiner[1] page by page. As shown in Fig. 2, an initial layout analysis is performed to organized character into a three-layer hierarical: LTTextLine l_{pdf}, NSCS n, LTChar c. Each LTTextLine consists of a list of LTChar and LTAnno as the separator. The NSCS is a sequence of LTChar separated by LTAnno. The value of a char plays an important role in the mathematical symbols checking. However, not all chars have a meaningful ASCII or Unicode value, as some PDF are with embedded CMAP font which is associated with a special mapping table. We use PDFbox[2] package to extract the font and the mapping table that maps a value to a readable glyph name, such as "bar", "leftbracket", and "delta".

3) Line Column Generation (LCG)

Document layout analysis aims to identify human perceived columns and lines, and it is crucial for ME identification [5]. The printed format of a ME is a 2D layout which might be split into multiple LTTextLines in PDF, especially for the IMEs. For academic publications, they are mostly in single or double column layout. For double-column pages, they might also have single-column header, footer or image/table on top. Based on this observation, we detect columns and lines using the concept of Projection Profiling Cutting (PPC) on the binary image I converted from PDF. The binary image uses value 1 for the black pixels and 0 for white pixels. A projection profiling (pp) is obtained by projecting the black pixels onto an axis and do a cumulative counting on each position on the axis. The horizontal and vertical pp for a PDF page are shown in Fig. 3.

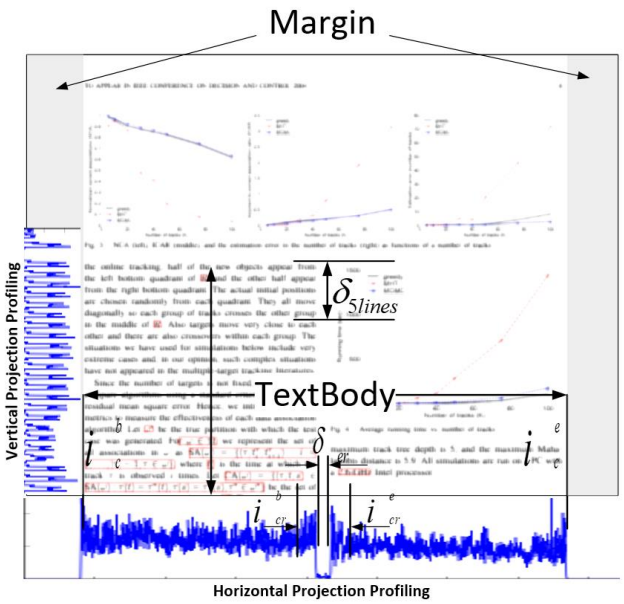

Figure 2. Illustration of layout analysis, 10.1.1.58.6850_6 in [13]

The two-step processing starts with double columns detection, followed by line segmentation. The detection of the double column format is based on a heuristic rule that there is at least 5 lines for the double column region between row pixel index i^{l}_{r} and i^{h}_{r}, s.t. $i^{h}_{r} - i^{l}_{r} > \delta_{5line}$, and there exist a central gap in the corresponding horizontal PP $pp_{h}(I[i^{l}_{r}: i^{h}_{r}, :])$.

The center gap is defined as an empty region of at least δ_{er} pixels around the center of the horizontal pp of text body region. We first get column range (i^{b}_{c}, i^{e}_{c}) of the text body by removing the empty margin. From the column range, we could estimate the central region (i^{b}_{cr}, i^{e}_{cr}), which is with a with that is in ratio α of the text body. Then we will look for δ_{er} consecutive empty pixels in the horizontal projection profile of the center region, $pp_{h}(I[i^{l}_{r}: i^{h}_{r}, i^{b}_{cr}: i^{e}_{cr}])$.

When the double column format is detected, one can find the largest $|i^{h}_{r} - i^{l}_{r}|$, $i^{l}_{r} < i^{h}_{r}$, with constraint that the horizontal PP of $I[i^{l}_{r}: i^{h}_{r}, :]$ has a central gap. Each column is passed to the line segmentation algorithm, which detects lines based on the zero gaps in the vertical pp.

By the end, for each LTTextLine l_{pdf} extracted from PDF file, we try to find a line region $L^{l}_{i,j,k}$ from PPC such that the overlapping area is at least half of the area of l_{pdf}. We enumerate through $\{L^{l}_{i,j,k}\}_{i,j}$ in page j of document D_i, and merge the associated l_{pdf} set to construct the lines.

The center gap ratio α is set to 0.1. The δ_{er} is set of 5. And δ_{5lines} is set empirically to 400 pixel. By manually checking the line detection results, we only failed at one case where there is an embedded figure. For the Isolated ME, the PPC based algorithm still has some errors in segment one under and over part of an IME into different lines.

1 https://github.com/euske/pdfminer
2 https://pdfbox.apache.org/

B. Heuristic rule to identify ME and NME

Common practices including styles, headings, etc. in different fields can be readily transformed into computable rules for content classification. The heuristic rules used in this paper are summarized as follows. For NME, different aspects

TABLE I. REGEX TO MATCH NON-ME ELEMENTS

Element type	Regex	Example
Citation	\[\d+(, \d+)*\] \((\D)*(18|19|20)\d\d\)	"[1, 17]", "(Tracy, 2000)"
Figure Table	(figure|fig.| table|tbl.|tab|form)[ ]* \d(\.\d)*[ ]*(\([a-zA-Z]\)|\[[a-zA-Z]\])	"Fig. 1a", "Tbl. 1.1"
Equation	(equation|eqn.|eq.|formula)[ ]* (\d+(\.\d+)*|\(\d+(\.\d+)*\))	Equation 1 Formula (9)
Theorem	(theorem|definition|example|corollary) [ ]*\d+(\.\d+)*	Theorem 1
Heading	(chapter|section)[ ]*\d+(\.\d+)*	Chapter 2 Section 2.2

of the technical work are organized into sections. Figures and tables and their captions summarize major results, and illustrate examples. Highlighted reserved words, e.g., theorem, lemma, etc. summarize key findings. MEs are composed of mathematical symbols and operators to represent real world objects and their system properties. They are also commonly used as a succinct expression of some repeatedly discussed abstractions. MEs can be treated as a form of text being blended with plaintext words into regular sentences. Some MEs may become reserved, *de facto* terminologies to represent sophisticated abstractions. Based on these heuristics, we propose several rules for the identification partial ME and NME characters at the levels of *symbol*, *NSCS*, *NSCS sequence*, and *line*.

Symbol level clues have very high precision in identifying ME. Based on Unicode value and glyph names, we can find out math symbols by checking whether they are Greek symbols, operators, relations, and big operators are identified [3]. As for line level, we can detect IME. If there is a symbol level clue or NSCS clue for ME, and there are no natural language words matched, we treat the line as an IME because plaintext words rarely appear in IME. The NSCS level and NSCS sequence level assessment are elaborated below.

1) NSCS level assessment

NSCS is a sequence of chars, essentially a string. If the string matches a known math function name or there exist any math symbols, we treat it as a ME word. As for NME detection, we try to determine whether the string matches plain text or is acronyms. For plain text matching, a string longer than 3 chars is normalized and then matched against the natural language corpus. The length thresholding technique aims to reduce the false positive rate. As for the word matching, we use Pattern.en [17] and Wordnet lemmatizer [18] to normalize noun or verb into its root form, which will then matched with the corpus of words provided by Natural Language Toolkit (NLTK) [15]. An acronym is typically formed from the first letter of multiple word sequence. We detect acronyms by checking the capitalization and the label of surrounding words.

2) NSCS sequence level assessment

Word sequence level is for the detection of reference, citation, and other document elements shown in Table I. Citation is for cross-document linkage [19]. We use regular expression (regex) to match APA style such as "(name, year)",

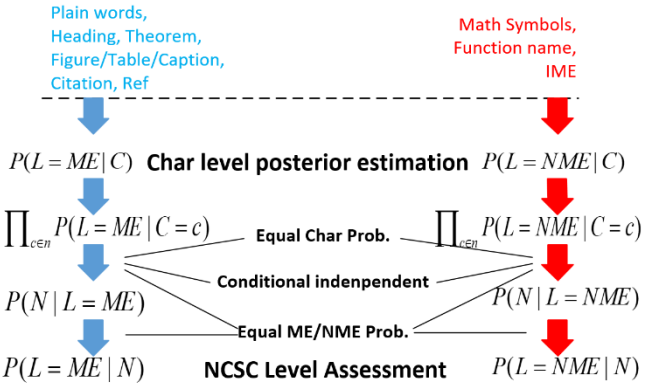

Figure 4. Workflow of the FSB model

"name (year)", and IEEE style such as "[index]". Intra-document element id and reference concern with figure, table, equation, block and heading. We use the grammar in [14] to design the regex to match the intra-document elements. Table I also shows the matching regex and examples. Unless otherwise stated, case is ignored when matching.

C. ME Extraction

At completion of the preprocessing steps, the FSB model can be used for ME extraction in two steps: FSB based NSCS level assessment; and EME merger based on NSCS labels and identified IME. The workflow for the FSB model is shown in Fig. 4. Through manipulating the conditional probability using Bayesian rules and conditional independence assumption, we obtain the key parameter. Then we will present the way to estimate the parameters and show an example. At last, we present the logic for EME merger.

1) The FSB Model

Let $F \in \mathcal{F}$ and $V \in \mathcal{V}$ be random variables of font and value defined over the char set $\mathcal{C}$. Let $N \in \mathcal{N}$ denote a random variable for a NSCS. The label $L \in \{ME, NME\}$ is defined over char set $\mathcal{C}$ and NSCS set $\mathcal{N}$. The goal of the FSB model is to label each NSCS $n \in \mathcal{N}$ as ME or NME. n will be labeled as ME when $L_R > 0.5$.

$$L_R(n) = \frac{P(L=ME|N=n)}{P(L=NME|N=n)} \quad (1)$$

Through a series of transformation, we will convert the above formula into what we can be estimated. Firstly, by applying Bayesian rule on $P(L|N)$, we have:

$$L_R(n) = \frac{P(N=n|L=ME)P(L=ME)}{P(N=n|L=NME)P(L=NME)} \quad (2)$$

Knowing that the combinatorial space for NSCS N is usually too large for probability estimation, we make the assumption of conditional independence here, where the $P(N|L)$ in (2) could be decomposed as follows:

$$\mathrm{P}(N = n|\mathrm{L}) = \prod_{\mathrm{c}\in n} \mathrm{P}(\mathrm{C} = \mathrm{c}|\mathrm{L}) \quad (3)$$

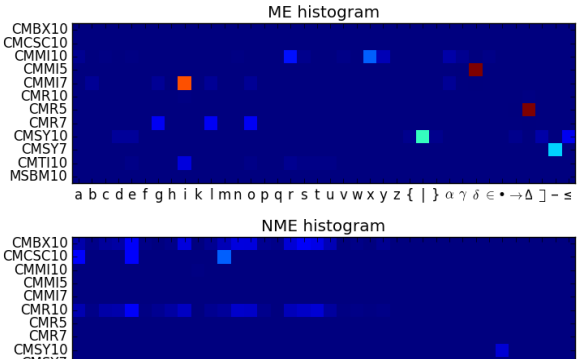

Figure 5. Normalized Font-Value co-occurrence matrix, from test file 10.1.1.6.2203_3 [13]

We further use the Bayesian rule to transform the likelihood $P(C|L)$ to posterior $P(L|C)$ to leverage on the discrimination power for ME/NME of font setting.

$$\mathrm{P}(N = n|\mathrm{L}) = \prod_{\mathrm{c}\in n} \mathrm{P}(\mathrm{L}|\mathrm{C} = \mathrm{c})\, P(C = c)/P(L) \quad (4)$$

We further make assumption of the equal probability of ME vs. NME, i.e., $P(L = ME) = P(L = NME)$. Then plug in (4) into (2), and cancel out $P(C = c)$, we have:

$$L_R(n) = \prod_{c\in n} \frac{\mathrm{P(L=ME\ |C=c)}}{\mathrm{P(L=NME\ |C=c)}} \quad (5)$$

This leads to the key parameter of posterior probability $P_c(L|C)$ of a character being a ME or NME for the NSCS level assessment.

2) $P_c(L|C)$ estimation

The estimation of conditional probability $P_c(L|C)$ is based on the font-value co-occurrence statistics in char set for C_{ME} and C_{NME} identified from the heuristic rules. We use matrix H_{ME} and H_{NME} for the font-value co-occurrence value, with each row for one font and each column for one value. We denote $H_{ME}(f, v)$ and $H_{NME}(f, v)$ as the count of co-occurrence of font f and value v for ME and NME. An example of the co-occurrence matrix is given in Fig. 5.

For good visual effects, we normalized co-occurrence as $H_*^n(f, v) = H_*^n(f, v)/H_t(f)$, where $H_t(f) = \sum_v H_{ME}(f, v) + H_{NME}(f, v)$ is the total amount of char in font f, and $*$ is ME or NME. As illustrated by this example, the font usage for ME and plaintext exhibit distinct patterns. The plain text is written in four major fonts, CMBX10, CMCSC10, CMR10, and CMTI10, while the ME are mainly written in the other fonts. But the font information is not enough to label a char as ME or not. The NME character, bullet, is in font CMSY10, where majority of characters in this font are ME, which show the necessity of using font-value pair to make assessment. Another challenge is observed from font CMTI10, where even the font-value pair could not determine whether the character as ME, which leads to our design of adding the NSCS level assessment to the FSB model.

Then we will estimate $P_c(L|C)$ based on two conditions. If the char $c \in C_{ME}$, we can estimate $P_c(L = ME|C)$ based on the font-value pair:

$$P_{fv}(L = ME|F = f, V = v) = \frac{\mathrm{H_{ME}(f,v)}}{\mathrm{H_{ME}(f,v)+H_{NME}(f,v)}} \quad (6)$$

Otherwise, we would like to estimate the probability using font information only, which would have more related samples to conduct estimation.

3) EME Merger

Through line level heuristic rules in III.B and conditional probability assessment model in III.C.1, we get a collection of predicted IMEs and NSCSs that are predicted as ME. For NSCSs not overlapping of any predicted IME, we merge consecutive NSCSs labeled as ME into an EME block.

IV. Experiment

A. Dataset and verification

Lin [13] provides a public dataset to evaluate ME extraction solution. For each ME, the annotation dataset includes the type of the ME (Isolated or Embedded), the bounding box of ME, and all characters/paths with associated bounding box. The bounding box is the smallest rectangle that contains the black pixels for the ME, which we call a tight bounding box. A total of 400 single-page files are labeled. However, the coordinate of characters by our PDF parsing library could not match the image provided in the dataset and we ignore them in the evaluation.

The first author of this work manually verify the ME labels and tallied a total of 1888 missing MEs. The missing annotation are mostly ME with less than five characters. MEs in figures are mostly not labeled, thus we also don't consider them and exclude them from the evaluation process. We keep and annotate the ME in caption and footnote as we observe that some of them are labeled in the original ground truth. As it is very hard to draw a tight bounding box in our annotation process, we have a further processing to shrink our annotation until a black pixel is met for both the ground truth and prediction.

B. Evaluation Criteria

We follow the criteria used in [6]. Since the EME is composed of NSCS, which consists of symbols, the assessment error at the NSCS level will inevitably affect the ME identification step. So we evaluate for both NSCS classification and ME extraction. For the NSCS classification evaluation, we use precision/recall/F1 as the criteria with ME as the positive samples.

The evaluation of ME extraction is based on the overlapping of bounding box (bbox) of the extracted MEs and the ground truth. The relation between a ME in ground truth

set $\mathcal{M}_{gt}$ and a ME in predicted set $\mathcal{M}_{pd}$ based on their bboxes could be: fully overlapping (OL), not overlapping at all (SP), contained (CT), or contain both overlapping and different part (OS). The overlapping is determined by a common area larger than 0.95 of both bboxes [13].

TABLE II. PERFORMANCE OF ANCHORING RULES

	Word	**Acro.**	**Cit.**	**Intra**	**Str.**	**Func**	**math**
ME	**68570**	**1147**	**687**	**998**	**416**	67	3849
NME	555	300	40	7	0	**190**	**26842**
Acc.	0.991	0.793	0.94	0.993	1	0.739	0.875

TABLE III. WORD LEVEL CLASSIFICATION CONFUSION MATRIX

		Prediction	
		NME	***ME***
Ground Truth	***NME***	124698	2776
	ME	3961	37779

TABLE IV. ME IDENTIFICATION RESULTS

	Cor	**Mis**	**Fal**	**Par**	**Exp**	**Pae**	**Mer**	**Spl**
IME	341	205	9	422	569	274	0	21
EME	633	953	1448	1383	4264	2952	10	10

A predicted ME m_{pd} could be one of the following type:

- Correct: $\exists\, m_{gt} \in \mathcal{M}_{gt}, rel(m_{gt}, m_{pd}) = OL$.
- Expanding: $\exists!\, m_{gt} \in \mathcal{M}_{gt}, rel(m_{gt}, m_{pd}) = CT$, where $\exists!$ is unique existential quantification.
- Merging: $|\{m_{gt} | rel(m_{gt}, m_{pd}) = CT\}| > 1$.
- Partial: $\exists\, m_{gt}, rel(m_{pd}, m_{gt}) = CT$, and no other $m_{pd}{}'$ satisfying this relation.
- Split: $\exists\, m_{gt}, rel(m_{pd}, m_{gt}) = CT$, and there are other m_{pd} with this relation, too.
- Partial and expand (PAE): a m_{pd} is labelled as PAE if it does not satisfy the above relation and $\exists m_{gt}, rel(m_{pd}, m_{gt}) \notin \{OL, SP\}$. This is different but more formal than the logic of the script given in [13] as their logic will miss some PAE cases.
- False: a predicted ME that is not matched with any relation mentioned above.
- Miss: a ME in ground truth that is not matched with any relation mentioned above.

The correctness requires an overlapping of bbox [13]. However, even if two character sets of a ME are the same (which is the correct case), it is hard to assert that their overlapping area is larger than 0.95. For example, the character 'i' is of size 21*40 in an image of size 4250*5500 converted from PDF file. One pixel changes on both horizontal and vertical dimension will result in an overlapping area of 0.92 only. This is also the reason for so many Expanding, Partial, and PAE cases in Table IV. But for an extracted ME and its corresponding ground truth, it is less likely to be not overlapping at all. This means the ground truth MEs labelled as Missing and extracted MEs labelled as False are reflecting the real situation. During the following discussion, we will use miss rate and false rate as the main indicator.

C. Results and Discussion

First, we would like to assess the performance of our heurstic rules. Then performance on the NSCS level classification is given. At last, we present results on the ME extraction.

1) Evaluation of Anchoring Rules

We show the effectiveness of our five rules for NME anchoring point detection and two rules for ME detection in table II. Each column represents a rule. The Intra-document elements include Figure/Table/Equation. The structure elements include Theorem and Heading. The first/second row is the number ME/NME NSCS identified by the rules. The third row is the accuracy according to the designed intention. Except for the abbreviation rule, the other four rules for NME detection are with high accuracy. The relative low performance of abbreviation might be due to the mixed used of abbreviation and the ME.

2) Performance for word classification

The samples of concern are the NSCS generated from our PDF processing toolkit. We would treat a NSCS n as a positive sample if its bounding box $B(n)$ overlaps with the bbox of a ground truth m_{gt} larger than a threshold, i.e.

$$\exists m_{gt} \in \mathcal{M}_{gt}, A\left(B(n) \cap B(m_{gt})\right) > \theta_o * A(B(n)), \quad (9)$$

, where the threshold θ_o is set to 0.1. The performance of NSCS-level classification is shown in the table III. The first/second row are normal text/ME NSCS. The first/second column shows the number of NSCSs predicted as normal text/ME samples.

We get 93.1% precision and 90.5% recall rate, 0.918 F1, while the best performance of word classification [6] as 88% precision, 82% recall, 83% F1. For the false positive part, a major cause is the incorrect labeling of the bounding box for some math symbols. It will result in a larger bounding box for an IME, which will include many plain text words, so that these plain text words will be treated as ME by the FSB model. For the false negative part, the errors are mostly numbers and parentheses. Another type of errors is because some words are indeed inside a ME, especially IME. The last type of false negative is due to inaccurate bounding box parsed from PDF such that some plain text words are overlapping with ME, which means their ground truth label is not correct.

3) ME identification

We evaluated the ME identification on IME and EME separately. Using a modified version of the script provided by [13], we get statistics on the number of perfect matches (Cor), miss (Mis), false (Fal), partial (Par), expanding (Exp), partial and expanding (Pae), merging (Mer), and split (Spl) as shown in table IV. We note that the correction number is not an accurate indicator because of the difference in the bounding box. The difference also causes the large portion of expanding ME. However, missing rate and false rate are not affected by the discrepancy in the bbox. These two criteria will be our main indicators for performance comparison.

The IME identification module is very simple and it achieve a satisfactory performance of 0.006 false rate and 0.111 miss rate. Our IME module could be compared with the line classification performance of previous work [6]. By converting the false rate and miss rate into precision/recall/F1 score, our system gets 0.994 precision and 0.889 recall, and a F1 score of 0.939. The first main cause of missed IME detection is that there are some common words for both math and plain text, such as “for”, “invariant”, “and”, “otherwise”, “super” in the IME. The other reason is the error in PDF processing. Some lines are failed to be extracted and the glyph name does not match the math symbols.

Our EME module is with a false rate of 0.135 and a miss rate of 0.093. In comparison with previous work [6], our system outperforms the best system with 5 percent better in miss rate and false rate. Besides, we add 1888 MEs during our verification process. The previous work might have worse miss rate when taken these extra MEs into consideration. The most common false case are the section numbers, reference to equation and some plain words connected with bracket. A particular case is a file with square brackets surrounding the reference. As for the missing part, single char variables are the common cause. The capitalized variables are also confused with acronyms.

4) Computational cost analysis

In summary, the average execution time (Python code based) for one page of pdf file is decomposed as follows: 1.89 seconds for layout analysis, 2.25 seconds for heuristic rule matching and font statistics, 0.22 seconds for IME identification, and 0.12 seconds for EME identification. This is much faster in comparison with supervised machine learning methods [6], which takes about 1 second to predict a line, 10 seconds to predict a word. It took 12 and 763 seconds to train line and word classifiers, respectively.

V. CONCLUSION

In this work, we propose a Font Setting based Bayesian (FSB) model to identify mathematical expressions by leveraging on knowledge about the natural language, technical publication practice, and probabilistic models. The complex task of ME extraction is involved with many processing steps for PDF parsing, document layout analysis and construction of resources. In addition to their applications for IME-EME identification and NSCS classification, the anchoring rules can be used as the building block for other high level applications. In comparison, the FSB model outperforms the leading ME extraction algorithm based on the machine learning method, in terms of the error rates and the processing time. FSB can be easily expanded to incorporate new rules or sophisticated filters, e.g., regex based matching.

ACKNOWLEDGEMENT

The toolchain prototype for this paper was made possible with the help of multiple open source projects. Their contributions to the community are greatly appreciated.

REFERENCE

[1] K.-F. Chan and D.-Y. Yeung, "Mathematical expression recognition: a survey," *International Journal on Document Analysis and Recognition*, vol. 3, no. 1, pp. 3-15, 2000.

[2] "Processing mathematical notation," in *Handbook of Document Image Processing and Recognition*, Springer London, 2014, pp. 679-702.

[3] X. Lin, L. Gao, Z. Tang, X. Lin and X. Hu, "Mathematical formula identification in PDF documents," in *2011 International Conference on Document Analysis and Recognition* , 2011.

[4] X. Lin, L. Gao, Z. Tang, X. Hu and X. Lin, "Identification of embedded mathematical formulas in PDF documents using SVM," in *IS&T/SPIE Electronic Imaging*, 2012, 2012.

[5] X. Lin, L. Gao, Z. Tang, J. Baker, M. Alkalai and V. Sorge, "A text line detection method for mathematical formula recognition," in *2013 12th International Conference on Document Analysis and Recognition*, 2013.

[6] X. Lin, L. Gao, Z. Tang, J. Baker and V. Sorge, "Mathematical formula identification and performance evaluation in PDF documents," *International Journal on Document Analysis and Recognition*, vol. 17, no. 3, pp. 239-255, 2014.

[7] R. Cattoni, "Geometric layout analysis techniques for document image understanding: a review," ITC-irst Technical Report 9703.09, 1998.

[8] M. Okamoto and B. Miao, "Recognition of mathematical expressions by using the layout structures of symbols," in *the First International Conference on Document Analysis and Recognition*, 1991.

[9] A. Kacem, A. Belaid and B. Ahmed, "Automatic extraction of printed mathematical formulas using fuzzy logic and propagation of context," *International Journal on Document Analysis and Recognition*, vol. 4, no. 2, pp. 97-108, 2001.

[10] B. Yu, X. Tian and W. Luo, "Extracting mathematical components directly from PDF documents for mathematical expression recognition and retrieval," in *International Conference in Swarm Intelligence*, 2014.

[11] J. Baker, A. Sexton and V. Sorge, "A linear grammar approach to mathematical formula recognition from PDF," in *International Conference on Intelligent Computer Mathematics*, 2009.

[12] J. Jin, X. Han and Q. Wang, "Mathematical Formulas Extraction," in *2003 International Conference on Document Analysis and Recognition*, 2003.

[13] X. Lin, L. Gao, Z. Tang, X. Lin and X. Hu, "Performance evaluation of mathematical formula identification.," in *10th IAPR International Workshop on Document Analysis Systems*, 2012.

[14] S. R. Choudhury, "Figure metadata extraction from digital documents," in *2013 12th International Conference on Document Analysis and Recognition*, 2013.

[15] S. Bird, E. Loper and E. Klein, Natural Language Processing with Python, O’Reilly Media Inc, 2009.

[16] "Comparing approaches to mathematical document analysis from PDF," in *2011 International Conference on Document Analysis and Recognition*, 2011.

[17] T. Smedt and W. Daelemans, "Pattern for python," *Journal of Machine Learning Research*, pp. 2063-2067, 2012.

[18] G. A. Miller, "WordNet: A Lexical Database for English," *Communications of the ACM*, vol. 38, no. 11, pp. 39-41, 1995.

[19] C. Lipson, Cite right: a quick guide to citation styles--MLA, APA, Chicago, the sciences, professions, and more, University of Chicago Press, 2011.