Detecting In-line Mathematical Expressions in Scientific Documents

Kenichi Iwatsuki
The University of Tokyo
7-3-1 Hongo
Bunkyo-ku, Tokyo 113-8656, Japan
iwatsuki@nii.ac.jp

Takeshi Sagara
InfoProto Co., Ltd.
1-15-2 Ochiai
Tama-shi, Tokyo 206-0033, Japan
sagara@info-proto.com

Tadayoshi Hara
InfoProto Co., Ltd.
1-15-2 Ochiai
Tama-shi, Tokyo 206-0033, Japan
harasan.g@gmail.com

Akiko Aizawa
National Institute of Informatics
2-1-2 Hitotsubashi
Chiyoda-ku, Tokyo 101-8430, Japan
aizawa@nii.ac.jp

ABSTRACT

One of the issues in extracting natural language sentences from PDF documents is the identification of non-textual elements in a sentence. In this paper, we report our preliminary results on the identification of in-line mathematical expressions. We first construct a manually annotated corpus and apply conditional random field (CRF) for the math-zone identification using both layout features, such as font types, and linguistic features, such as context n-grams, obtained from PDF documents. Although our method is naive and uses a small amount of annotated training data, our method achieved an 88.95% F-measure compared with 22.81% for existing math OCR software.

KEYWORDS

PDF structure analysis; mathematical formula recognition; in-line mathematical expression detection; math IR; scientific paper mining

1 INTRODUCTION

Recent advances in natural language processing (NLP) techniques have enabled researchers to automatically extract and use scientific knowledge from scientific publications [6, 11]. However, these techniques require "natural language sentences" that can be analyzed by standard NLP tools, such as dependency parsers or named entity extractors. Extracting natural language sentences from PDF documents is quite laborious and time-consuming and thus causes a serious delay in the development of semantic analysis.

In the past, the automatic extraction of metadata has been the main motivator in PDF structure analysis [10, 13]. Several works also exist that target the identification of the section structures of scientific papers [3, 12], and extraction of figures and tables [2]. However, the quality of the extracted sentences, in terms of NLP, has not been sufficiently considered in previous studies.

One of the critical issues in extracting natural language sentences from PDF documents is the identification of non-textual elements in a sentence. In this paper, we postulate that identifying *in-line* mathematical expressions is specifically important. In-line mathematical expressions include complex mathematical structures, such as "$\sum$" or "$\int$", in addition to symbols or variables that accompany their explicit natural language definitions, such as "where w is a sequence of words" or "the probability distribution $p(W|c)$." Unexpectedly, our analysis showed that it is not a trivial task to distinguish in-line mathematical expressions from dictionary words that appear in italics. Identifying this notation is useful not only for reducing the errors of sentence parsing, but also enabling further scientific text mining because mathematical expressions often convey key concepts in scientific information dissemination.

Recently, deep-learning techniques have just created an opportunity for progress on mathematical OCR, which is the task of converting mathematical images into MathML [4], which is a standard XML representation of mathematical expressions[1]. Once such a technique is established, identifying in-line mathematical expressions becomes more critical. For example, the definitions of mathematical variables can effectively enhance the search of independent-line mathematical expressions [7]. However, to the best of our knowledge, there has never been a study that has focused on the detection of in-line mathematical expressions.

Based on this background, we report in this paper our preliminary results on the issue. In our method, we first construct a manually annotated corpus and apply conditional random fields (CRFs) for math-zone identification using both layout features such as font types, and linguistic features such as context n-grams, obtained from PDF documents. We investigate the effectiveness of the method by comparing the results with an existing math OCR tool (InftyReader[2] [5]). We also identify influential features by an ablation test to demonstrate that both types of features contribute to performance. Although our method is naive and uses only a

[1] https://www.w3.org/Math/
[2] http://www.inftyproject.org/en/

DocEng'17, September 4–7, 2017, Valletta, Malta

ACM ISBN 978-1-4503-4689-4/17/09…$15.00
https://doi.org/10.1145/3103010.3121041

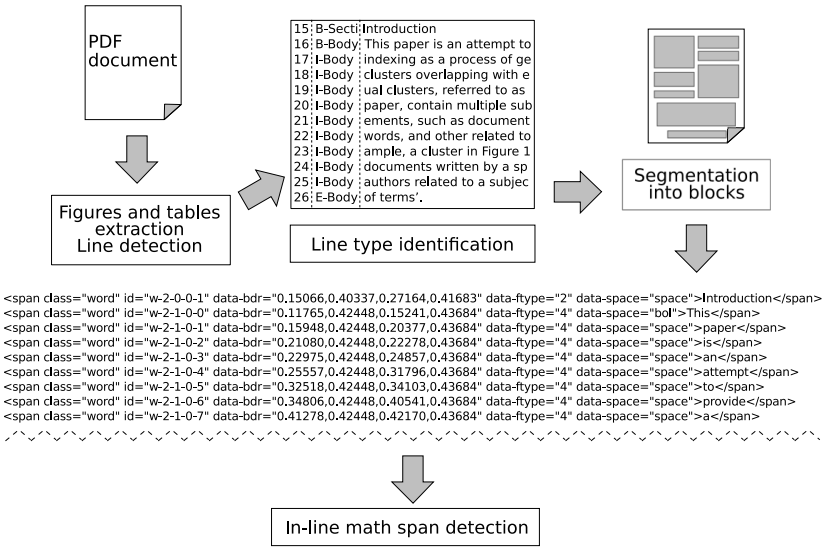

Figure 1: Preparing data for the in-line math span detection module.

small amount of annotated training data, the results are promising and we expect that our investigation will provide a strong basis for similar techniques.

2 RELATED WORK

2.1 PDF Structure Analysis of Scholarly Documents

Despite the worldwide movement toward XML publication by leading publishers, the majority of scientific papers are still published ing PDF format [11]. Since PDF is a layout-based format for printing, it is not easy to recognize the logical structure of a paper if only PDF files are available.

Existing tools to analyze PDF scientific articles have mostly applied machine learning techniques, such as support vector machines (SVMs) or CRF, to manage the ambiguity. Several state-of-the-art tools include CERMINE (SVM and CRF) [13], GROBID (CRF) [10], and OCR++(CRF) [12]. In a comparative study in 2015 [9] GROBID was reported to be the best performing system for metadata extraction. Then, CERMINE and OCR++ were reported to be improvements over GROBID.

Evaluation datasets has been created manually [12] based on the bibliographical meta-data that corresponds to PDF papers [9, 10] or semi-automatically constructed using the XML full-text version of papers [13].

2.2 Formula Identification

Non-textual objects in scientific papers include figures, tables, and pseudo-code. Among them, mathematical expressions that appear as in-line objects are difficult to recognize using standard layout analysis techniques.

Recently, Lin et al. [8] proposed a method for identifying a text line for independent-line mathematical formulae. Once the regions for mathematical expressions are identified, automatic formula identification methods can be applied to convert the math images to standard formats such as MathML. Deng et al. [4] applied constitutional and recurrent neural networks for the image-to-XML conversion of mathematical expressions and reported 75% accuracy on their dataset. InftyReader [5] can identify in-line math expressions. However, because only limited features are considered for the detection, it may not be sufficient to handle a wide range of mathematical expressions. In this paper, we use a more general framework based on sequential labeling.

3 METHOD

3.1 Data Preparation

First, we needed to determine where we should place our in-line math detection module within the workflow of PDF structure analysis. We assumed that in-line math detection was applied after segment identification and labeling, but before sentence splitting (Figure 1). Therefore, the input to the detection module was a sequence of space-separated tokens with their font types and the positions of the corresponding word bounding boxes. Linguistic features, such as POS tags and dependencies, were not considered in this study.

Considering the availability of manually annotated labels, we used our own in-house PDF analysis tool [1] for data preparation. In our workflow, all the text lines in the document were first extracted using poppler[3], ImageMagick[4], and pdffigures [5]. Using information about the positions of characters, the utility called pdftotext included in poppler automatically integrated characters into words. Then, we manually categorized the text lines into 38 predefined classes including *title*, *author*, *section header*, *theorem*, and *figures*. Then, consecutive text lines with the same label were merged into a single block. The blocks that were labeled as *body* became the input to the in-line math detector.

[3]https://poppler.freedesktop.org/
[4]https://legacy.imagemagick.org/
[5]http://pdffigures.allenai.org/

It should be noted that, unlike up-to-date tools such as GROBID, our annotation explicitly labeled independent-line math formulae, which enabled us to exclude them from our dataset. We used manually annotated data only for the purpose of evaluation. Automatic segmentation was also possible, for which we obtained the accuracy 96.54% with CRF when we used 130 papers as training data. For this paper, we focused on born-digital PDF papers, not OCR-processed PDFs.

3.2 Construction of the Annotated Dataset

We selected 74 papers from ACL Anthology[6]. Each paper contained a minumum of one mathematical expression, a maximum of 699 mathematical expressions, and an average of 156 mathematical expressions. The average number of words per math span was 3.61. Of all the math spans, 11.3% consisted of only one letter, which usually denoted a variable and whose detection was useful for tasks such as the extraction of the explanation of each variant.

In our annotation, each maximal text region of the mathematical expression that satisfied the following policy was annotated as an "in-line mathematical expression." For further details, refer to the guidelines at https://github.com/Alab-NII/inlinemath.

(1) Annotate every minimal region that (structurally and semantically) satisfies the following conditions:
(1-a) the region can be recognized as playing the role of (structurally/semantically) composing a natural language sentence containing the region; and
(1-b) the region has a mathematically-closed (independent) structure.
(2) Cover any structures and notations specific to mathematics, which do not appear in the text of neutral domains.

3.3 CRF Features

To identify in-line mathematical expressions, we tagged every word using CRF[7]. We used three types of labels: beginning of math expression (B), inside math expression (I), and outside math expression (O). The size of the window around the current word was five words (two words to the left and two words to the right). For CRF features, we used the following:

- `word` (word)
- `font` (font of the word (font name and font size))
- `length` (length of the word)
- `samew` (whether the same word is contained in an independent formula) (T/F)
- `samef` (whether the same font is used in an independent formula) (T/F)
- `samewf` (`samew` and `samef`) (T/F)
- `alpha` (whether the word consists of only letters) (T/F)
- `greek` (whether the word contains a Greek letter) (T/F)
- `math` (whether the word contains a mathematical symbol) (T/F)
- `single` (whether the word is a single character) (T/F)
- `mainfont` (whether the font of the word is the same as the body text) (T/F)
- `block` (label of the block containing the word)
- `url` (whether the word is a part of a URL (T/F)

[6]http://aclanthology.info/
[7]We used CRFsuite (http://www.chokkan.org/software/crfsuite/) in our implementation.

4 EXPERIMENTS

4.1 Baseline and Performance Measures

As a baseline, we used the latest version of InftyReader[8] Ver.3.1.3.1. InftyReader [5] is currently the most commonly used OCR-based mathematical content recognition system. For preprocessing, InftyReader returns the positions and sizes of the rectangular regions of all mathematical expressions in a document. Based on the information, we identified the in-line mathematical expressions detected by InftyReader.

The performance was evaluated using precision, recall, and the F-measure calculated as follows: $Precision = \frac{a_i}{b_i}$, $Recall = \frac{a_i}{c_i}$, and $F\text{-}measure = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$, where a_i denotes the number of words in paper i that are math expressions and tagged correctly, b_i denotes the number of words that are tagged as math expressions, and c_i denotes the actual number of words in the math expressions.

4.2 Performance

The precision, recall, and F-measure values for five-fold cross-validation are shown in Table 1. *Micro-average* denotes the case in which the score was calculated based on the word count. *Macro-average* denotes the average of the performance scores for each paper. As is shown in Table 1, InftyReader achieved a high recall but low precision, which indicates that it selected candidates with low confidence. By contrast, our CRF-based method performed reasonably for both precision and recall, which resulted in a substantial performance difference in terms of F-measure (22.81% vs. 88.95% for the micro-average, and 23.84% vs. 80.41% for the macro-average).

To determine which feature was important, an ablation test was conducted. The same experiments were conducted after removing one feature. If the removed feature was significant, the scores should have decreased a great deal. Table 2 shows the results of the ablation test. The `word` and `mainfont` features were the two most influential features, which demonstrates that both linguistic (e.g. `word`) and layout (e.g. `font`) features contributed to the identification task.

Table 1: Overall performance.

	Precision	Recall	F-measure
InftyReader (micro-avr.)	13.06%	89.98%	22.81%
Our method (micro-avr.)	94.93%	83.68%	88.95%
InftyReader (macro-avr.)	13.82%	86.74%	23.84%
Our method (macro-avr.)	86.54%	75.09%	80.41%

4.3 Error Analysis

We further analyzed the cause of the relatively low recall values. Our analysis demonstrated that the problematic math expressions could be categorized into the following three types: The first type is mathematical expressions infrequently used, such as V_{todo},

[8]http://www.sciaccess.net/en/InftyReader/

Table 2: Results of feature ablation test (micro average).

Removed feature	Precision	Recall	F-measure
word	88.77%	73.99%	80.71%
font	93.54%	77.21%	84.60%
length	92.67%	78.25%	84.85%
samew	92.02%	78.98%	85.00%
samef	92.69%	79.35%	85.50%
samewf	92.69%	79.36%	85.51%
alpha	93.09%	78.92%	85.42%
greek	92.42%	79.10%	85.24%
math	92.96%	79.13%	85.49%
single	92.71%	89.98%	85.59%
mainfont	92.67%	77.42%	84.36%
block	93.16%	77.36%	84.52%
url	92.81%	79.52%	85.65%

$RankPos_i$, and lp_{misc}, which are not typically contained in learning datasets. Since word features are significant, these expressions are unlikely to be classified as mathematical expressions. The second type is parentheses. There were some cases where the system recognized "(" as a math expression but not ")." The third type is variables in texts. We found many failure cases where variables were expressed using the same font as narrative text, such as "p", not "p." Since the font feature was weighted highly in our trained CRF, variables with body text fonts were unlikely to be correctly recognized.

4.4 Discussion

Table 3 shows the relationship between the number of papers used for training. It can be observed that the amount of performance improvement was only small compared with the data size. Table 4 shows the top 10 weighted features obtained using the CRFSuite default options. Our CRF relied more on individual words than general features and seemed to overfit the training set. A simple solution for this would be to increase the size of the training dataset. However, considering the annotation cost, a more realistic solution would be to use more elaborate feature engineering or introduce a neural network-based framework.

Table 3: Relationship between the number of learned papers and scores (micro-avr.).

# of learn data	37	59.2	72	73
# of test data	37	14.8	2	1
Precision	91.56%	94.93%	92.66%	92.54%
Recall	78.61%	83.68%	80.51%	80.75%
F-measure	84.60%	88.95%	86.16%	86.25%

5 CONCLUSION

We proposed a method to detect in-line mathematical expressions using CRF and achieved an 88.95% F-measure, even though the learning dataset was small. To the best of our knowledge, this is the first work that has focused on in-line math, and the score was far better than the existing math OCR tool. We found that words and fonts were important for distinguishing math expressions from narrative text.

Table 4: Top-10 weighted features.

Feature (1-5)	Lbl.	Wght.	Feature (6-10)	L.	W.
word[0]=.	O	3.17	word[0]=M2	O	-2.04
word[0]=,	O	3.12	word[0]=M1	O	-1.95
word[0]=CW1	B	2.46	word[0]=M1	B	1.79
word[0]=CW2	B	2.25	word[0]=)	O	1.67
word[0]=M2	B	2.14	mainfont[0]=T	O	1.56

Detecting in-line mathematical expressions is important to exploit information from scholarly articles using NLP technologies. Note that narrative text and math expressions should be addressed in a different manner. For example, it matters whether "a" is a variable or an indefinite article. The proposed method also serves as a critical component in math information retrieval for scientific articles. For example, we can apply the description extraction developed in [7] to generate pairs such as {W, "*a sequence of words*"} and {A_{ji}, "*acoustic frames*"}.

The CRF-based implementation is simple and can be easily applied to a large scale scientific paper collection. We expect that the proposed method would be the baseline for in-line math detection and related areas.

ACKNOWLEDGMENTS

This work was supported by JST CREST Grant Number JPMJCR1513, Japan.

REFERENCES

[1] Takeshi Abekawa and Akiko Aizawa. 2016. SideNoter: Scholarly Paper Browsing System based on PDF Restructuring and Text Annotation. In *COLING*. 136–140.
[2] Christopher Clark and Santosh Divvala. 2016. PDFFigures 2.0: Mining Figures from Research Papers. In *JCDL*. 143–152.
[3] Isaac G. Councill, C. Lee Giles, and Min-Yen Kan. 2008. ParsCit: an Open-source CRF Reference String Parsing Package. In *LREC*. 661–667.
[4] Yuntian Deng, Anssi Kanervisto, and Alexander M Rush. 2016. What You Get Is What You See: A Visual Markup Decompiler. *arXiv preprint arXiv:1609.04938* (2016).
[5] Yuko Eto and Masakazu Suzuki. 2001. Mathematical Formula Recognition Using Virtual Link Network. In *ICDAR*. 762–767.
[6] Halil Kilicoglu. 2017. Biomedical Text Mining for Research Rigor and Integrity: Tasks, Challenges, Directions. *Brief. Bioinform.* (2017).
[7] Giovanni Yoko Kristianto, Goran Topic, and Akiko Aizawa. 2017. Utilizing Dependency Relationships between Math Expressions in Math IR. *Inf. Retr. J.* 20, 2 (2017), 132–167.
[8] Xiaoyan Lin, Liangcai Gao, Zhi Tang, Josef Baker, Mohamed Alkalai, and Volker Sorge. 2013. A Text Line Detection Method for Mathematical Formula Recognition. In *ICDAR*. 339–343.
[9] Mario Lipinski, Kevin Yao, Corinna Breitinger, Joeran Beel, and Bela Gipp. 2013. Evaluation of Header Metadata Extraction Approaches and Tools for Scientific PDF Documents. In *JCDL*. 385–386.
[10] Patrice Lopez. 2009. GROBID: Combining Automatic Bibliographic Data Recognition and Term Extraction for Scholarship Publications. In *TPDL*. 473–474.
[11] Horacio Saggion and Francesco Ronzano. 2016. Natural Language Processing for Intelligent Access to Scientific Information. http://taln.upf.edu/pages/coling2016tutorial/COLING2016_T3_NLP_FOR_SCIENTIFIC_PUBLICATION_v7.pdf. In *COLING*. 9–13.
[12] Mayank Singh, Barnopriyo Barua, Priyank Palod, Manvi Garg, Sidhartha Satapathy, Samuel Bushi, Kumar Ayush, Krishna Sai Rohith, Tulasi Gamidi, Pawan Goyal, and Animesh Mukherjee. 2016. OCR++: A Robust Framework For Information Extraction from Scholarly Articles. In *COLING*. 3390–3400.
[13] Dominika Tkaczyk, Paweł Szostek, Mateusz Fedoryszak, Piotr Jan Dendek, and Łukasz Bolikowski. 2015. CERMINE: Automatic Extraction of Structured Metadata from Scientific Literature. *Int. J. Doc. Anal. Recognit.* 18, 4 (2015), 317–335.