Mathematical Expression Retrieval in PDFs from the Web Using Mathematical Term Queries

Kuniko Yamada[1] and Harumi Murakami[1]

Osaka City University, 3-3-138, Sugimoto, Sumiyoshi, Osaka 558-8585 Japan
d18ud512@eb.osaka-cu.ac.jp

Abstract. Since mathematical expressions on the web are not annotated with natural language, searching for expressions by conventional search engines is difficult. Our method performs web searches using a mathematical term as a query and extracts expressions related to it from the obtained PDF files. We convert the PDF to TeX, create images from the mathematical descriptions in TeX and obtain image feature quantities. The expressions are discriminated by a support vector machine (SVM) using the feature quantities. Our experimental results show that eliminating slide-derived PDF files effectively improves F-measure and the mean reciprocal rank (MRR) is best when using both PDFs and HTML.

Keywords: Mathematical expression retrieval · PDF file · Web search.

1 Introduction

There is much useful mathematical information on the web, especially in PDF files that contain reliable information. However, it is difficult to search for mathematical expressions in PDF files efficiently. Our method performs an ordinary text search using a mathematical term as a query and presents mathematical expressions related to an input query from the PDF files. After converting the PDF to TeX, we create images from the mathematical descriptions in TeX and obtain the image feature quantities. Our method measures the relevance between a query and a mathematical expression from the following viewpoints: the expression is in a separate line; the query is in the neighborhood and it has appropriate image feature quantities when converting it to an image; and it appears in the first part of the PDF file.

2 Approach

2.1 Mathematical expression

Mathematical documents have features which differ from ordinary documents. The writing style for mathematical expressions in a document is: (a) variables and signs, or an additional expression are on a line and are expressed with math

notations instead of characters because the mathematical font is special; (b) an important theorem or formula is on a separate line, even though it is part of a sentence; and (c) when a theorem is derived or a calculation example is shown, its expression is very long. In such a case, the expression is usually not important.

2.2 Method

Our method is illustrated in Fig. 1. This approach does not depend on the language; however, we conducted our original study in Japanese, so we translated the data into English for ease of explanation. In our previous study [1], we proposed a method which retrieves mathematical expression images on the web using mathematical terms as queries. Here, we propose a method which retrieves mathematical expressions in PDF files. In Fig. 1, highlighted parts ((2) and (3)) are the new proposal and the others are adjusted to suit the TeX documents.

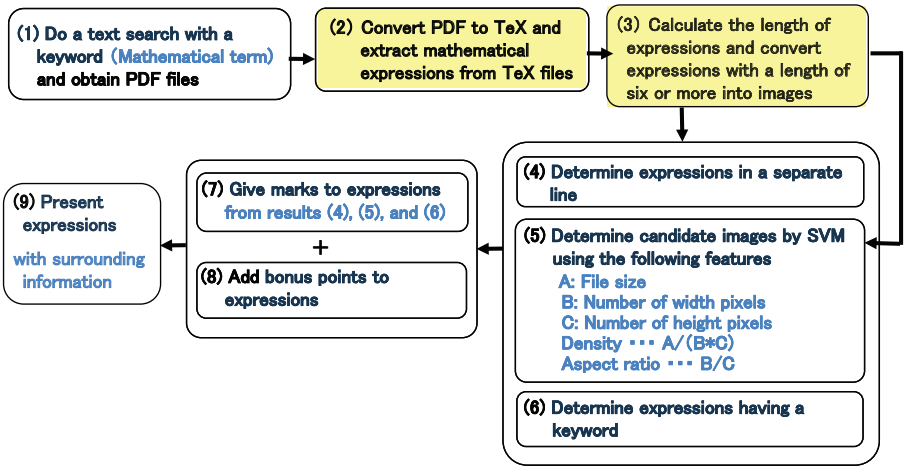


Fig. 1. Overview

(1) We perform a web search using mathematical terms as a query and capture the top 100. From the search result page returned by the search engine, we obtain PDF files and the boldface parts of the snippets.

(2) Preliminary experiment results showed that PDF files derived from slides had poor accuracy. In order to eliminate these, we obtain the aspect ratio of the PDF files from "MediaBox" and "Rotate" in the PDF page object, and remove landscape-oriented files beforehand. Then, we covert the PDF files to TeX files using InftyReader (http://www.inftyproject.org/en/index.html).

(3) In order to eliminate variables and fragments of expressions, we define and calculate the length of a mathematical expression according to the following procedure. (a) TeX commands are extracted from dataset D_1 (described later)

and the command is given a 1 if it is a constituent element of an expression; otherwise, it is given a 0. For example, commands with 1 are "\frac", "\partial", and "\alpha", and commands with 0 are "\mathrm"and "\overline". (b) These TeX commands are checked to determine whether they match part of the expression. The number of commands with 1 is length1 and all matched commands are deleted. The number of the characters in the rest of the expression is length2. The length of the expression is then the sum of length1 and length2. Since the length of $y = f(x)$ or $E = mc^2$ (E=mc^2 in TeX notation) is 6, we define 6 or more as a mathematical expression.

(4) When a mathematical expression part in TeX is in the display math mode, the expression is given a 1. When an expression part is in the inline mode, the expression is given a 0.

(5) Mathematical expressions converted to images are classified by SVM. We convert an expression to a PNG format image and use the SVM classifier developed by LIBSVM [2] from dataset D_2 (described later). We previously developed this classifier in [1]. The feature quantities are shown in Fig. 1(5). If an output of the SVM is positive, the expression is given a 1 and, if not, a 0.

(6) We search for the keyword (query) as a window size that is set from -200 to +200 characters surrounding a mathematical expression in TeX. If it exists, the expression is given a 1; if not, a 0. However, when there is no expression with a keyword, we search again for an alternative keyword which is the boldface part of the snippet.

(7) Each expression is given a score using (4) to (6). At this point, three points is a full score.

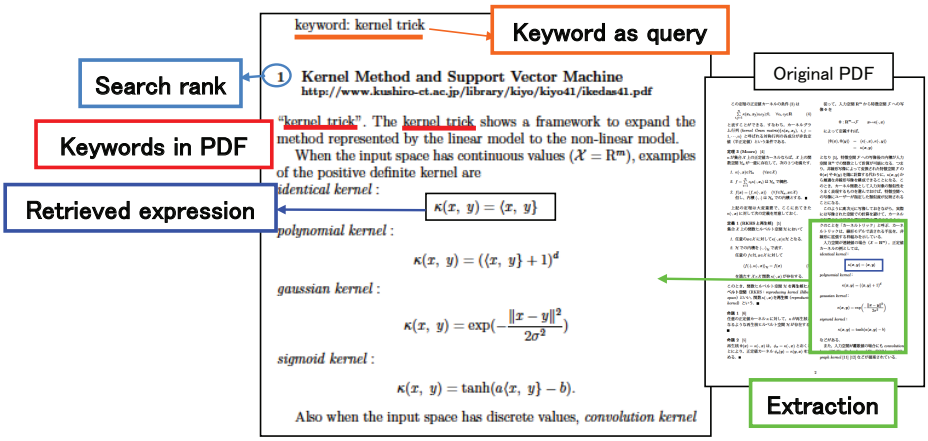
(8) The first order of expressions is returned by the search engine and the appearance order in the PDF file. When expressions are ranked in descending order of their score, incorrect expressions are still highly ranked because their order is affected by the ranking of the original web pages. As important information often appears in the first part of a document, we propose a bonus point system to solve this problem. This system gives an additional point to an expression when it first obtains the full score in the appearance order in the PDF. With this system, the possibility increases that the correct expression in each PDF rises to a higher order. From the above, $score(i_k)$ is given to each expression i_k by Eq. (1) in [1].

$$score(i_k) = x_{line} + x_{key} + x_{svm} + x_{bo}, \tag{1}$$

when $x_{line} = 1$, an expression is in a separate line; when $x_{key} = 1$, an expression has a keyword; when $x_{svm} = 1$, an expression is discriminated to be positive by the SVM; and when $x_{bo} = 1$, an expression has a bonus point.

(9) Our system ranks expressions according to $score(i_k)$, and obtains the top ten. Finally, the top ten expressions with their surrounding information are displayed. In order to show collateral conditions and commentary on the expressions, 200 characters before and after the expression are also displayed. Figure 2 shows a screen display example. Here, the top one with "kernel trick" as the keyword was retrieved correctly. This expression is extracted from the

end of the second page of the original five-page PDF document. This PDF ranks 12th in the original ranking returned by the search engine.



Some right parentheses changed to brackets as a result of conversion mistakes.

Fig. 2. Display example of keyword: kernel trick

3 Experiments

3.1 Dataset

In [1], we randomly selected keywords from the index of "Pattern Recognition and Machine Learning [3] (Japanese version)" by Bishop and conducted a web search using keywords as queries. We obtained the top 100 results for each keyword. We created dataset D_1 using keywords 1 to 30, D_2 using keywords 31 to 60, and D_3 usingis keywords 61 to 70 for evaluation. Table 1 shows the contents of D_3. The expressions in D_3 were judged manually.

Table 1. Contents of dataset D_3

	PDF			HTML			
	Expression		File	Image		File	Other
	Correct	Total		Correct	Total		
Total	398	29,708	297 (included 60 slide-derived PDFs)	181	6,030	554	149

Keywords: positive definite matrix, multi modality, equality constraint, Newton-Raphson method, root-mean-square error, Lagrange multiplier, sum rule of probability, kernel trick, uniform sampling, kinetic energy
Other: includes broken links, other file formats, PDF that cannot be converted to TeX, etc

3.2 Experiment1

The aim of this experiment was to examine the effect of eliminating slide-derived PDF files. Using F-measure, MRR (Eq. (2)), and mean average precision (MAP), we evaluated the output results of the top ten. MAP was obtained by calculating the macro mean of the Average Precision (AP) (Eq. (2)).

$$\mathrm{F-measure} = \frac{2 \cdot \mathrm{Precision} \cdot \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}, \quad \mathrm{MRR} = \frac{1}{n}\sum_{k=1}^{n}\frac{1}{r'_k}, \quad \mathrm{AP} = \frac{1}{\min(n,c)}\sum_{s} I(s)\mathrm{Prec}(s), \tag{2}$$

where $\mathrm{Precision} = \frac{r}{n}, \mathrm{Recall} = \frac{r}{c}$, r is the number of correct expressions of top n, c is the total number of the correct expressions, r'_k is the rank of the correct expression at the top of the k-th keyword, $I(s)$ is the flag indicating whether the expression at s-th is correct or not, and $\mathrm{Prec}(s)$ is precision at s-th.

In Condition A, our method is applied to all PDF files in D_3 and, in Condition B, it is applied to the PDF files which were not derived from slides in D_3. The results show that the F-measure and MRR improved by eliminating slide-derived PDF files, as illustrated in Fig. 3 and Table 2.

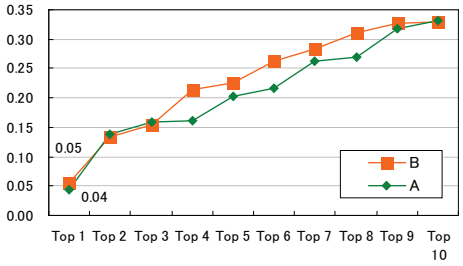


Fig. 3. Results of Experiment1 (F-measure)

Table 2. Results of Experiment1 (MRR, MAP)

	A	B
MRR	0.50	**0.56**
MAP	**0.17**	**0.17**

3.3 Experiment2

From the results of Experiment 1, we used the PDF files with the slide-derived PDFs eliminated. The aim of this experiment was to compare three conditions: Condition B uses PDF (same as in Experiment 1); C uses HTML (same as in [1]); and D uses both PDF and HTML, and these are ranked again after restoring to the original ranking by the search engine. As shown in Fig. 4 and Table 3, almost all values demonstrate that C is better than B and D; however D is the best in MRR.

3.4 Discussion

PDF files have many correct expressions (see Table 1) and clear sources. 99.4% of D_3 was obtained from organizations such as universities, academic societies,

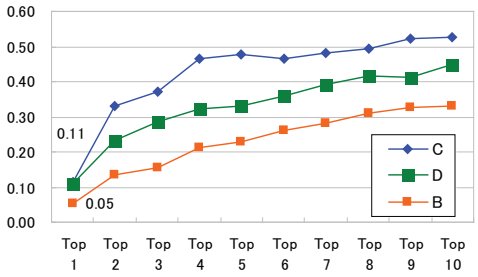


Fig. 4. Results of Experiment2 (F-measure)

Table 3. Results of Experiment2 (MRR, MAP)

	B	C	D
MRR	0.56	0.75	**0.77**
MAP	0.17	**0.34**	0.29

research institutes, and enterprises. To use these PDF files more effectively, our future tasks are to: (1) increase evaluation data to verify experimental results in more detail; (2) develop a new classifier to improve accuracy; and (3) perform preprocessing, other than the length, to improve accuracy.

4 Related work

Mathematical expression retrieval usually means retrieval of similar mathematical expressions and a search scope within a particular database. For example, Zanibbi et al. conduct similar mathematical expression retrieval which searches in the dataset developed from arXiv using mathematical expressions written in LaTeX and those converted into images [4]. We were unable to find any research using a dataset from PDFs on the web.

5 Conclusions

We proposed a method to perform web searches using a mathematical term as a query and extracted mathematical expressions related to it from the obtained PDF. Our experiments demonstrate the usefulness of using PDFs other than slide-derived PDFs as well as using both PDF and HTML in MRR.

References

1. Yamada, K., Ueda, H., Murakami, H., and Oka, I.: Mathematical expression image retrieval on web using mathematical terms as queries. Transactions of the Japanese Society for Artificial Intelligence **33**(4), A-H91_1–13 (2018)
2. Chang, C.-C. and Lin, C.-J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1–27:27 (2011)
3. Bishop, C. M.: Pattern Recognition and Machine Learning. Springer-Verlag, New York (2006)
4. Zanibbi, R. and Yuan, B.: Keyword and image-based retrieval for mathematical expressions. In: Proceedings of Document Recognition and Retrieval XVIII, pp. 011–019 (2011)