Vol.2017-IFAT-125 No.11
Vol.2017-DC-104 No.11
2017/3/10

A Structure-based Method for Mathematical Document Classification

Tokinori Suzuki[1,a)] Atsushi Fujii[1]

Abstract: Mathematical document classification (MDC) is a task to classify mathematical documents consisting of text and mathematical expressions (ME) to mathematical categories, e.g. probability theory and set theory. This task is an important task for supporting user search on recent wide-spreaded digital libraries and archiving services. Although ME could bring an important information as being in a central part of communication especially in math fields, how to utilize ME for MDC is not matured. In this paper, we propose the classification method based on texts combined with structures of ME, which are expected to reflect mathematical concepts and rules specific to a category. We demonstrate classification results that our proposed method outperforms existing method with state-of-the-art ME modeling on F-measure.

Keywords: Mathematical Document Classification, Mathematical Information Retrieval, Tree kernel SVM

1. Introduction

A mathematical document is a document subjected to mathematical communication such as a math paper and discussion in online Q&A community. *Mathematical document classification (MDC)* is a task to classify mathematical documents (Figure 1 A) to mathematical categories, e.g. probability theory and set theory. This task is an important task for supporting user search on recent wide-spreaded digital libraries and archiving services.

The fundamental difference between an ordinal text classification and MDC is on that documents contain *mathematical expressions (ME)* in the body. ME forms a main line of communication. For example, discussion in a math paper is usually carried along with ME for disambiguating concepts explanation written in natural languages. ME is a combination of symbols, that is well-formed according to the grammar and rules specific to mathematical categories. Therefore, how to utilize the information of ME is an important technique for MDC.

To the best of our knowledge, a few of studies have addressed MDC by simple adaptation of "Bag-of-words"; by using only textual features in [17] or by textual features with symbol frequency in ME [3], [21]. In relation to ME modeling which is applicable to MDC, the modeling methods have been studied on a *mathematical search (MS)* task, which is a task to search mathematical expressions/documents with ME and/or keywords.

MS has been studied actively in this decade as seen in a major task in NTCIR conferences [1], [2], [24]. Most of the devotion of previous works [5], [6], [7], [10], [20] has been put into the superficial match of ME, because of the task setting to find out the close ME on the same appearance by queried ME (introduced in Section 2). Due to the difference of objective of MS and MDC, just adapting the ME modeling to the MDC would suffer from a following problem.

The problem comes from an aspect of ME, that is highly symbolized and sometimes even one symbol can represent concrete mathematical concepts. For example, Figure 1 A) shows two documents in group theory and graph theory. Both documents share a particular symbol. The group theory document contains a symbol "G" for representing the concept of a group. On the other hand, "G" is for the concept of a graph. Even though the character means different concepts, the existing modeling possibly links those similar to each other on the appearance, which would make the classification worse.

To solve the problem, our idea is leveraging structures of ME possibly carry category specific information such as conventions of ME, symbols usages, function names and so on. Here, a structure of ME is formed by the MathML markup language in Figure 1 B, which corresponds to the tree structure in Figure 1 C. An investigation on structures of ME in graph theory and group theory is summarized in Figure 2. The figure lists up the most frequenct subtrees which contain symbol "G" (same character and different semantics) and are constituted by layout tags: <mrow>, <msub> and <mover>. Picking up the <mrow> subtrees, $V(G)$ and $E(G)$ trees are the majorities in graph, which mean functions that return sets of vertices and edges of a graph. While, $Z(G)$ and $H^n(G;Z)$ are the most part of mrow in group, which mean a center of a group and a cyclic group. In this way, structures represents the difference of symbol usages.

To make use of the structures, we propose a classification method with the structural kernel method [14], which can automatically generate the effective features in tree structured objects. We also propose techniques to overcome the limitation on expressibility of MathML tree (discuss in Section 3) using MathML

[1] Tokyo Institute of Technology, Tokyo, Japan
[a)] suzuki.t.co@m.titech.ac.jp

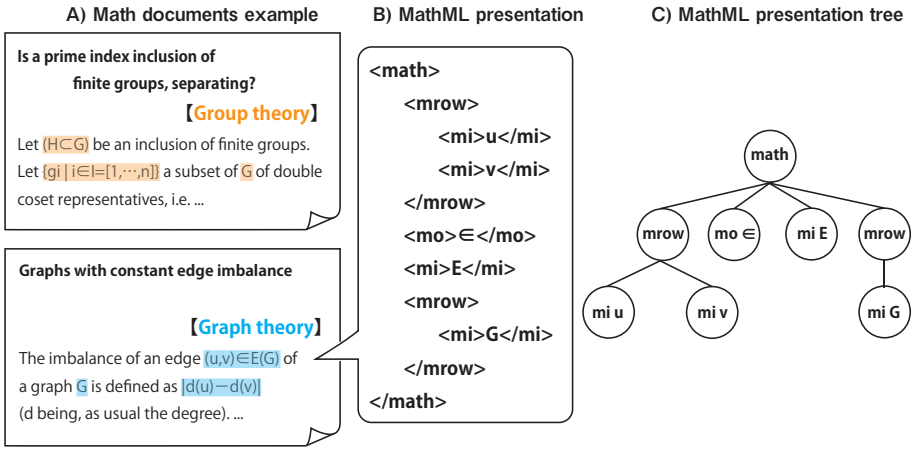

Fig. 1 Example of mathematical documents consist of text and mathematical expressions (ME) in shaded parts, one of ME in MathML and the correponded tree to the ME

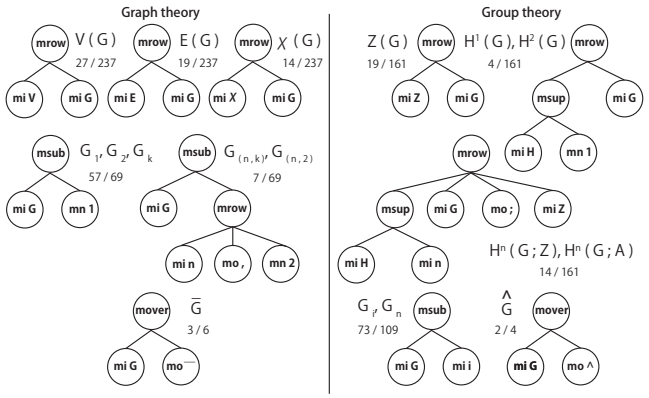

Fig. 2 Subtrees frequently occur in each Graph theroy and Group theory of a Q&A community site. These subtrees include symbol "G" and are formed by layout nodes: mrow, msub and mover.

tag and frequent substructures of category when applying the tree kernel method to ME trees.

A primary contribution on this paper is taht our proposed method outperforms the classification with state-of-art ME modeling by about 3 on F-measure.

This paper is outlined as follows. Section 2 mentions the related works. Section 3 presents our proposed method to utilize structures of ME. Section 4 introduces document collections for MDC evaluation. Section 5 reports experiments and the results and we discuss the results in Section 6. Finally Section 7 draws conclusions.

2. Related work

This paper presents MDC methods with use of a structural kernel method and developing collections for MDC evaluation. In this section, we introduce the existing works in relation to mathematical search, the structural kernel method and test collections for Math IR.

2.1 Math Expression Modeling on Mathematical Search

Mathematical search (MS), which is a task to search for ME or mathematical documents by querying with ME/keywords of text, has been studied actively in this decade. For ME representations, there are several markup languages, such as LaTeX, MathML[*1] and OpenMath[*2]. MathML presentation format (Figure 1) is the major markup on scientific articles on the web, which carries simple layout information of symbols. In this paper, because it is common and widely used markup, we use MathML as the standard format of ME. MS method can be categorized two types as to ME modelings: *text-based*, *structure*.

First category is incorporating MS into the text retrieval framework. *Bag-of-symbols (BOS)* has been proposed by adapting the well-known *bag-of-words* model to MS [13], [15], [16]. BOS models index mathematical symbols in ME. Given a ME "$(u, v) \in E(G)$" in Figure 1 B, "u" and "∈" are the one of tokens. *N-gram* model [13] makes indices of N-gram of the symbols. [15], [16] augment the symbols with a mathematical feature which is the type of the symbol. In Figure 1 B, the index maintains the symbol "*u*" with a type "mi" in their index. The index also holds the pairs of an operator-argument as a combination feature. As text-based alternatives, linear sequencialzing approaches [8], [12] flatten a ME to a LaTeX form e.g. "$x^{t-2} = 1$" to "x^{t-2} = 1" and serch the ME over the linearized ME by text matching.

Second category, *structure* model utilizes the structural information of ME. This modeling is closely related to this work on the point that our method also use structural information of ME. The category can be two directions: Indices for efficient search on ME tree [5], [6] and indices for tree-based similarity search [7], [10], [20].

Since the number of ME in target documents is huge, the efficiency of search is of interest in MS. Two works [5], [6] proposed indices to support efficient search of ME. Substitution tree index [5] is a tree structured index where they map substituiton rule of subexpression of ME to the path of the tree index. The other index [6] of subtrees in MathML is also proposed for efficient access of ME by boolean retrieval. Althoguh these works concern with ME structure, main focus is on looking up ME by an exact match or boolean retrieval.

For tree-based similarity search, one indexes math symbols along with positional information of the symbols in ME tree [20],

[*1] http://www.w3.org/Math/
[*2] http://www.openmath.org/

such as depth of a node for prioritizing the local matching in math tree. Tree edit distance [7] is applied to MS for measuring similarity among a query and the target ME in tree structure, which tends to search ME with high similarity of whole a tree structure rather than of local part of ME tree like due to the influence of cost of edit on tree. Operator tree [10] which is a tree constructed by reflecting the priority of operators into the tree level is proposed for normalizing ME such as "$1+2*3$" equals "$2*3+1$". *Pattern* model [9] is proposed to search based on sub-expressions such as $E(G)$ by manually defined templates with wild cards, such as "$*(*)$".

2.2 Structural Kernel Method

As far as related fileds to this work, structural kernels have been applied to natural language processing tasks. Structural kernels are effective means to extract features automatically in natural language texts. In kernel machines, learning and classification algorithms depend on the ability to compute similarity score; $\sum_{i=0}^{n} \omega_i K(x_i, x) + b$, where $K(x_i, x)$ is a kernel function defines mapping from objects to feature vectors. In case of strctual kernel K determines the shape of the substrcute of the objects. Several kernel have been proposed, for example, string kernel [19], syntactic tree kernel[4] and partial tree kernel [14].

In particular, tree kernel method has been explored in retrieving question and answer task [18] over the NLP fundamental tasks recently, which successfully extracting structural relationship of shallow parse tree (syntax, POS and so on) among question and answer. However, regarding to MDC, the possibility of applying tree kernel to the task has never been studied.

3. Classification Method with Substructure of Math Expression

As discussed in Section 1, structures of ME can carry some of the rules of using symbols and the conventions, e.g. function names, symbol usages and writing styles, specific to the category. In order to take advantage of the above information for the classification, we employ supervised learning approach to the classifying where a classifier is used to learn a model to discriminate one specific math category from the other categories with tree kernel SVM [14].

Tree kernel SVM can automatically extracts and learns discriminative features in tree structured objects. To adapt the classification to the tree kernel approach, single document d is represented by as a pair of MathML trees T of ME and a normal feature vector v of text in a document, namely, $d = (T, v)$. Given two documents d_i and d_j, we define the following kernel:

$$K(d_i, d_j) = TK(T_i, T_j) + K_v(v_i, v_j)$$

where TK computes a tree kernel similarity between ME represented in MathML tree. K_v is the kernel over feature vectors of the text. A tree kernel on MathML tree TK is defined as follows:

$$TK(T_1, T_2) = \sum_{n_1 \in NT_1} \sum_{n_2 \in NT_2} \Delta(n_1, n_2)$$

where NT_1 and NT_2 are the sets of the nodes in T_1 and T_2. $\Delta(n_1, n_2)$ is equal to the number of common fragments rooted in the n_1 and n_2 nodes, which are subtrees. We propose three approaches based on above the kernel function.

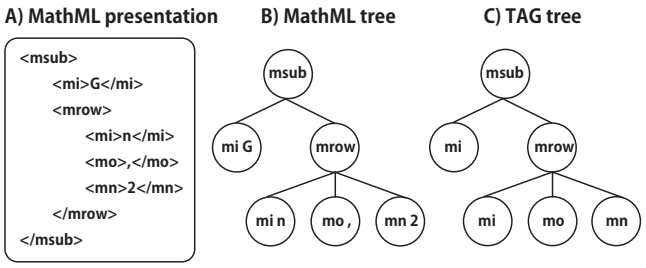

Fig. 3 Example of MathML tree and MathML tag tree of a math expression $G_{(n,2)}$

3.1 MathML tree approach

First approach uses MathML tree in Figure 3 only for the tree kernel similarity (TK). This approach is a simple but effective for in the domain of mathematical documents. Because we have to handle the vast forest of ME in math documents(discussed in the statistic part of Section 4) to classification, tree kernel makes us avoid expensive feature engineering of trees of math expressions.

3.2 Using MathML tag tree

Second approach is to add MathML tag (TAG) tree to each tree for smoothing . In Figure 3, $G(n, 2)$ is a common notation in graph theory which is a definition of a graph indicating the number of vertices and edges of the graph. Including parameters (in this case n and 2) in its body, ME of a graph definition could be variable such as $G(6, 2)$ and $G(8, 3)$ or $G(n, i)$ and $G(n, k)$.

Even though those expressions could be conceptually close in terms of the graph definition and a representative feature for classifying the category, simply adapting tree kernel method to MathML tree would suffer from a superficial issue caused by parameters. Since the fragments generated by tree kernel method are different from each other due to the parameter difference, the method cannot link the notation to the specific category.

To alleviate the problem, we propose TAG approach to link the fragments. In addition to MathML tree, this approach adds a tree of MathML tag of ME (the right side of Figure 3). The tree comprises nodes of the tag set of MathML. In Figure 3, for example, one of the subtree (mrow (mi)(mo)(mn)) generated from a part of MathML tree (mrow (mi n)(mo ,)(mn 2)) is able to match the other-generated subtree such as (mrow (mi n)(mo ,)(mn 7)).

3.3 Frequent substructure of math expression

Third approach makes use of frequent sub-graph of ME (FRE), which aims to extract category specific features. We assume that a category specific feature is a well-used function, a operator and its argument and notations expectedly reflecting writing of ME unique to the category. For example, "$H^3(G; Z_{(p)}) = H^3(S; Z_{(p)})N_G(S)/C_G(S)$" is a equation in a group theory document in Figure 4. This equation contains frequently used a group notation "$H^*(G; *)$" (* means a wild card) which colud be discriminative for group theory.

To take the structural clue to the classification, the substructure have to be extracted, a side from auto-generated subtree features

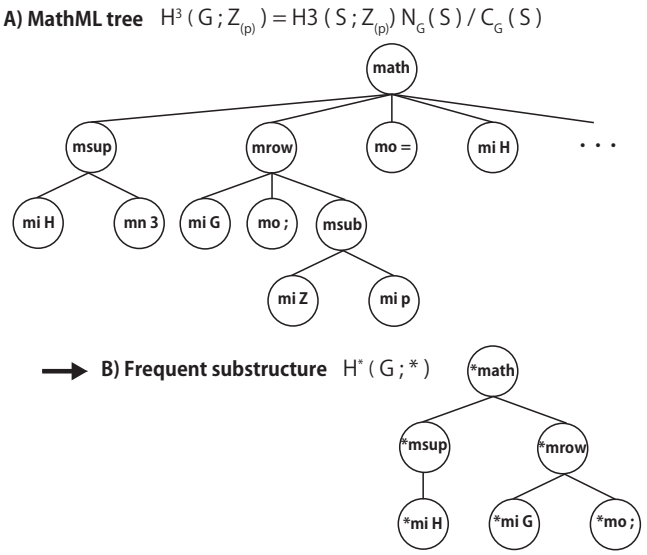

Fig. 4 Example of frequent structure of a math expression $H^3(G; Z_{(p)}) = H^3(S; Z_{(p)})N_G(S)/C_G(S)$

by tree kernel method. The substructure could be any subgraphs of MathML tree. Simply adapting tree kernel method cannot exploit the substructure due to a MathML limitation. MathML define simple symbol placement to form ME, which cannot express the priority of ME in tree structure such as the an argument of a function should be descendant of the function. For example, the arguments $G; *$ should follow H^* as a descendant in the tree for the priority.

Therefore, we compute frequent substructures in a category with following steps:

(1) Pool all the math expressions in the documents in a category.
(2) List up frequent subgraphs of the math expression trees in the category by a frequent subgraph mining algorithm.
(3) Omit some of the subgraphs which consist of less than three nodes and do not contain both operator node <mo> and identifier <mi> or numerical literal node <mn>.

In step 2, we use gSpan algorithm [22] for frequent subgraph mining. The algorithm requires the minimum number of frequency for target subgraphs, which is a parameter of this approach. We manually decide for each category. Next in step 3, what we object to capture is category specific subexpression, for example, the writing of function and the relationship operator and variable specific to the category. For the purpose, the subgraph with only two nodes or consisting of only one type of node could not carry the information. That leads to do the preprocessing.

4. Test Collection for Math Document Categorization

In this section, we present the test collections used for MDC evaluation, category settings and some basic statistics on documents and categories.

4.1 Mathematical Catgory & Document

First of all, we used Math subject classification scheme *(MSC2010)* [*3] as a category system for this evaluation. MSC2010 is a classification scheme formulated by editorial staffs of Mathematical reviews, which is a journal of the American Mathematical Society, in order to help users find the papers of interest to them as readily as possible. Within the category, we built test collections by crawling the *MathOverflow*[*4] pages[*5] and *arXiv*[*6] papers:

MathOverflow is the most active Q&A community being updated constantly on mathematical subjects targetting professional mathematicians. A document in mathoverflow collection is a pair of a question and a series of answers. For a categorial setting, we took advantage of user tags assigned to questions. The user tags comprise not only keywords in mathematics but mathematics categories which can be found in MSC 2010 top and second-level math subjects. We selected principal 15 categorial tags on which the community user actively discussed. For each category, around 300 documents are gathered. This collection is balanced on the aspect of document number. number. Table 1 shows the details of the categories and the number of documents. Mathoverflow collection consists of 3 339 documents in total.

arXiv is the digital library of scientific papers in various fields, e.g. mathematics, physics and so on. A document in this collection corresponds to one paper submitted in arXiv math field. Paper types vary from a short abstract paper to a long journal paper. Although arxiv math papers are also tagged categorial labels, the category system is slightly different from MSC 2010 because of a differential of purposes. The arxiv category system is set by the cordinators in arxiv math field, which is designed for user convinience of accessing papers. arxiv's "Quantum Algebra", for example, does not correspond to any counterparts in MSC 2010 due to sum up the other topics into one. Twenty three categories out of total thirty two arxiv categories correspond to MSC 2010 top and second-level math subjects as well as the mathoverflow collection.

Then, we collected all the papers submitted into arxiv math subject in 2014 (# of papers is 37 735). Table 2 shows a part of the documents used in this evaluation, which are successfully converted its latex-coded ME to MathML with a converting tool MathToWeb[*7]. The number of documents on each category ranges from roughly 100 to 2 000 because of a biased popurality of papers submission in math fileds.

4.2 Statistic of Collection

In the following, we investigate the basic statistics of the collections regarding document contents, thus text and math expressions (ME) and categories.

Dividing document contents to text and ME, we investigate statistics of the both text and ME. Table 3 shows the number of both words and ME in the collections. The average numbers of words are about 600 in mathoverflow and 4 700 in arxiv. In arxiv collection, the number fluctuates between 105 at a minimum and 64 090 at a maximum because of paper types: a short abstract paper to a long journal paper. The average number of ME is 28 and

[*3] http://www.msc2010.org

[*4] http://www.mathoverflow.net
[*5] Mathoverflow contents are available under the Creative Commons Attribution Share Alike (CC-BY-SA)license
[*6] http://arxiv.org/
[*7] http://www.mathtoweb.com

Table 1 Math category and the number of documents in MathOverflow collection

ID	Category	# of documents	ID	Category	# of documents
1	Algebraic geometry	222	9	Representation theory	256
2	Number theory	267	10	Category theory	258
3	Combinatorics	285	11	Commutative algebra	254
4	Algebraic topology	247	12	Linear algebra	277
5	Group theory	259	13	Logic	288
6	Differential geometry	270	14	Set theory	327
7	Probability	271	15	Graph theory	279
8	Functional analysis	293		Total # of documents	3 339

Table 2 Math category and the number of documents in arXiv collection

ID	Category	# of documents	ID	Category	# of documents
1	Commutative algebra	376	13	Logic	391
2	Algebraic geometry	1 300	14	Metric geometry	403
3	Algebraic topology	401	15	Numerical analysis	857
4	Combinatorics	2 358	16	Number theory	1 302
5	Category theory	186	17	Operator algebras	369
6	Complex variables	511	18	Probability	1 570
7	Differential geometry	1 153	19	Rings & algebras	551
8	Dynamical systems	999	20	Representation theory	636
9	Functional analysis	920	21	Symplectic geometry	226
10	General topology	140	22	Spectral theory	276
11	Group theory	753	23	Statistics theory	517
12	Information theory	1 288		Total # of documents	14 384

Table 3 Statistics of textual words and the mathematical expressions in a document

	# of words			# of ME		
Collection	Min	Avg	Max	Min	Avg	Max
MathOverflow	15	630	6 428	1	28.3	374
arXiv	104	4 713	64 090	1	363.2	5 061

Table 4 Statistics of trees of mathematical expressions

	Node			Height		
Collection	Min	Avg	Max	Min	Avg	Max
MathOverflow	1	9.72	171	1	2.58	12
arXiv	1	13.68	149	1	2.80	19

Table 5 Proportion of documents with multiple categories in each category

Collection	Min	Avg	Max
MathOverflow	0.205	0.381	0.691
arXiv	0.114	0.468	0.741

363 in mathoverflow and arxiv respectively.

Since ME are represented in a tree structure, we also investigate the tree statistic in Table 4. The average height of ME tree in both collectons are close to each other at about 2.5, which is quite low. The reason could be that many of the documents contain one-symbol ME and it decreased the average. On the number of nodes, that of arxiv is the more than that of mathoverflow reflecting long equations in papers.

We also examined the categorial overlaps of the documents. Figures in Table 5 are percentages of documents in multi categories on each category. On average, 38% of mathoverflow and 46% of arxiv documents are with multiple categories.

5. Experiment

In this section, we conduct experiments to compare classification methods with the test collections in Section 4. We mention the experimental set-up, evaluation metrics and the results.

Table 6 Method and the feature (TF means term frequency)

Method	Description
BOW	TF of a word in text part of a document
BOS	TF of a symbol in an ME (in [15])
CO	Combination of 1-3 gram of a ME (in [15])
OT	TF of subtrees of ME operator trees (in [10])
TK	Tree kernel SVM to MathML (in Section 3)
TAG	Tree kernel SVM to TAG tree(in Section 3)
FRE	Tree kernel SVM to FRE tree (in Section 3)

5.1 Experimental Set-up & Evaluation

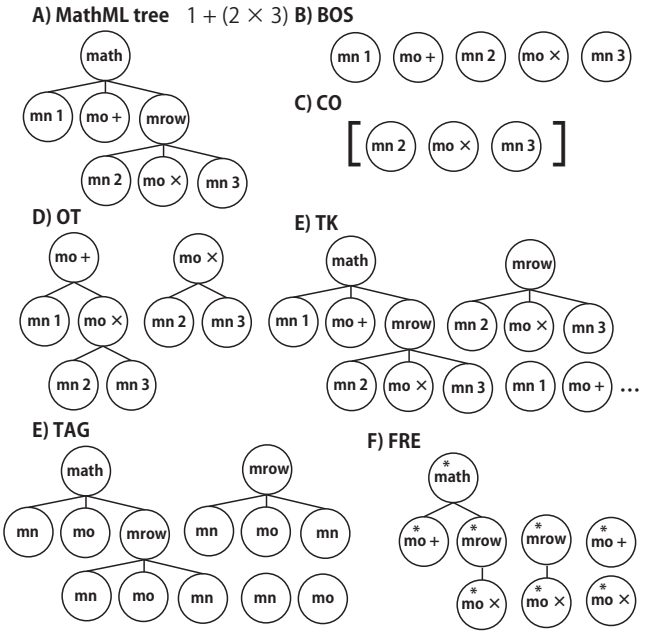

Fig. 5 Example of features and tree kernel generated fragments of 1+(2×3)

To evaluate calssifications, we compare seven methods combined with approaches in Table 6 (feature examples in Figure 5). BOW feature is *term frequency (TF)* of the words in text part of

Table 7 Examples of extracted structures of ME in group theory

Structures	Tree	Source	Meaning
$H^*(G;$	(math (msup (mi H))(mrow (mi G)(mo ;)))	$H^n(G;A)$	A part of the definition of a subgroup H
$:G \rightarrow$	(math (mo :)(mi G)(mo →))	$f:G \rightarrow Q$	A part of map function among two groups
$\|G\|$	(math (mo —)(mi G)(mo —))	$\|G\| = \|H\|$	The index of a group G
(x, y)	(math (msup (mi H)(mrow (mi G)(mo ;))))	$(x, y) \in \mathbb{Q} \times \mathbb{Q}$	A pair of values

a document after stop words removal with SMART system stop list[*8]. BOS feature is TF of symbols in ME with a math feature and CO is the combinations (uni-tri gram) of ME symbols [15]. OT feature is the TF of subtrees of operator trees of ME [10]. TK, TAG and FRE are proposed methods utilizing tree kernel method to ME in MathML introduced in Section 3. TAG method adds TK with trees consisting of MathML tags. FRE method includes category specific frequent substructures of ME. For example, Table 7 shows the example of extracted substructures of ME with high frequency in group theory. Successfully, our extraction technique could extract some of the fragments denote a subgroup definition and a map function. Though, the technique extracted the substructures like (x, y) which occurs frequently in not only group theory but in the others because of common expressions in general mathematics.

Since classifications with features modeled by TF showed the higher classfication performance than the classifications with the features modeled by TF-IDF on preliminary experiments, we use TF as features in both text and ME for classifications.

For classification evaluation, we evaluate a binary classifier which sorts out one specific category and the rest. Thus, we calculate precision, recall and F-measure on a single category as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \ \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{F}_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

where TP (true positive) is the number of documents assigned to the category correctly. FP (false positive) is the number of documents assigned to the incorrect category. FN (false negative) is the number of documents that belong to the category but which are failed to be classified to the category. F-measure is calculated by the harmonic mean of precision and recall.

5.2 Result: Classification performance

Table 8 lists the outcome of experiments, where we compared seven methods on one-vs-rest classification setting. The figures are the average of precision, recall and F-score in mathoverflow and arxiv categories on 10-fold cross validation setting.

On the mathoverflow result, one of the proposed methods (ID 7) showed F-score at 71, that is the highest in the methods, improving the precision value compared to text-only feature by about 8 points. On recall side, the symbol-based existing methods (ID 2,3) indicated the higher recall at around 73 than the others, while the precision of the methods are bit lower than the others.

On the arxiv result, the ID 7 method achieved the best F-score as same as the result of mathoverflow. Though, the tendency of the precision and recall indicated by the methods is quite different from in the other collection. In this collection, the highest precision is marked by BOW method at 69 and proposed three methods (5,6,7) follow the figure by less than 1 point. Regarding recall, the symbol-based methods also show the highest recall while the precision at 49 value dropped clearly from that of the other methods at around 68.

For detailed result of method, Figure 6 shows the three effective measures of each method in 15 categories of mathoverflow and 23 categories of arxiv. Overall, there is not a constant improvement by proposed methods over categories but a certain improvement on some categories, that raises the classification performance in both datasets. For the methods (5 and 6), the recall lines project on category 9 (representation theory) and 7 (probability) in Figure 6 A. For the method (7), the diagram are basically outside of that of the others on precision improving especially on category 8 (functional analysis) and 12 (linear algebra).

In Figure 6 B, the precision graph shows that symbol-based method 2 and operator-tree method 4 are obviously the lower on the value than the others spreading the charts inside the other lines on most of the category. While BOW achieved the best precision over the categories, the recall is slightly lower than the others because the value of some of categories such as category 5 (category theory), 10 (general topology) 19 (rings&algebras) and 22 (Spectral theory) decreased the average by 10 points from the top of the recall on each category. As to proposed methods 6 and 7, there are clear improvements on category 9 (functionl analysis) and 16 (number theory) on precision by increasing the precision 15 points from BOW method.

5.3 Result analysis

For result analysis, we conduct significance test in order to determine the category where our method can improve the classification, since our proposed method showed the best classification performance among the methods, though, F-score is not so different from that of BOW. We, then, study individual documents in the category where significance is observed.

We conducted two-sided paired T-test on both mathoverflow and arxiv results to check whether there is a mean difference between the proposed methods and existing methods on each category. Table 9 presents the test result on mathoverflow results. In Table 9, the table header denotes pairs of methods. For example, "5" over "1" means the pair of the method 5 and 1 (These method numbers correspond to method ID in Table 8).

On a few of mathoverflow cateogies, statistical differences could be observed for the method 5 (BOW+TK) method as showed in Table 9, where only two categories, Probability and Representation theory, that show the mean difference between the method 1 (BOW). More categories are with significance for the

[*8] http://jmlr.csail.mit.edu/papers/volume5/lewis04a/a11-smart-stop-list/english.stop

Table 8 Average performance of each method over all the categories (Figuers in %)

ID	Method	MathOverflow collection			arXiv collection		
		P	R	F	P	R	F
1	BOW	64.784	71.921	68.166	69.418	71.008	70.204
2	BOW + BOS	63.745	73.453	68.255	49.252	78.742	60.606
3	BOW + BOS + CO	63.712	73.353	68.194	49.244	79.074	60.692
4	BOW + OT	66.288	71.924	68.991	62.141	77.723	69.064
5	BOW + TK	66.602	73.255	69.770	68.658	74.987	71.683
6	BOW + TK + TAG	68.597	73.100	70.777	68.661	74.468	71.447
7	BOW + TK + FRE	72.201	69.838	71.000	68.762	75.868	72.141

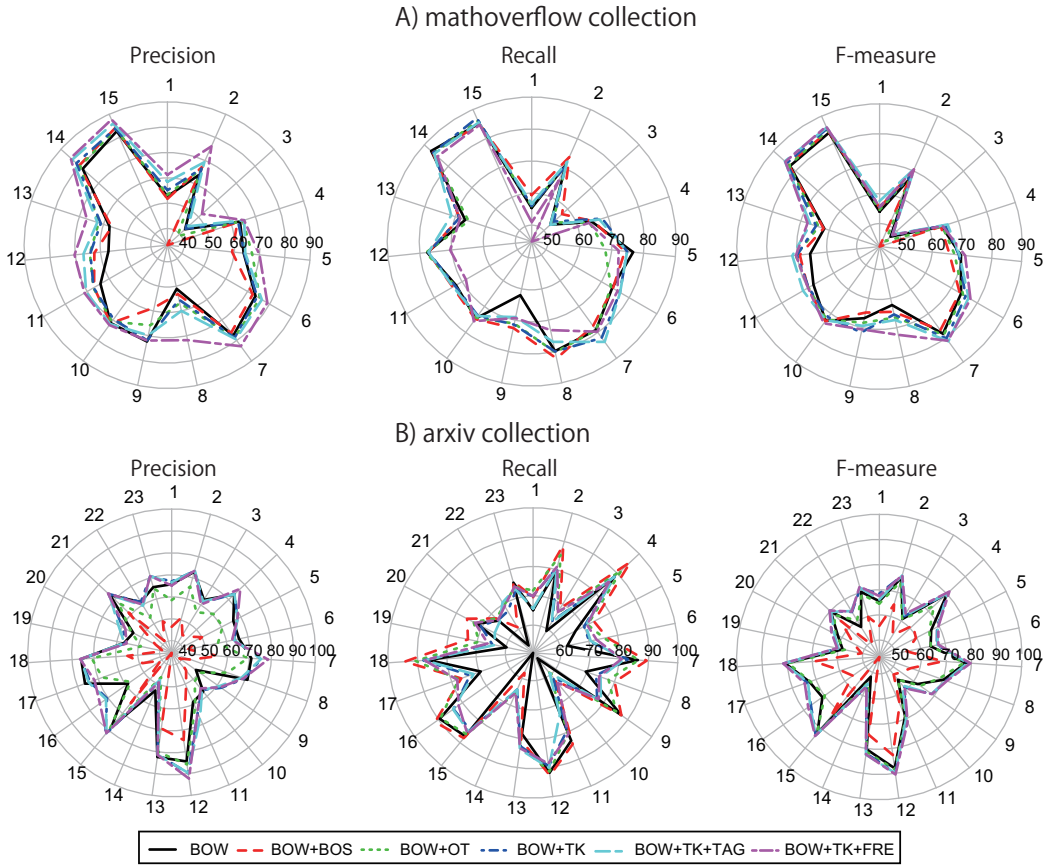

Fig. 6 Precision, recall and F-measure of the methods on each category (category ID outside of circles corresponds to ID in Table 1 for Mathoverflow and Table 2 for arXiv)

method 6 and 7 than the method 5.

In terms of arxiv results, on the category 2 (algebraic geometry), 4 (combinatorics), 8 (dynamical systems) and 18 (probability) the method 5, 6 and 7 are without significance to method 1 (BOW). Except for these comparisons, three proposed methods show the significance to exisiting methods on every category at $p < 0.01$ level.

Next, we investigated individual documents in the category with the significance. On probability, the method 5 (BOW+TK) is able to classify other six documents and where the existing methods fail to classify. Picking up remarkable examples of those, two documents would have the other main topic rather than probability and even both of the documents contain several ME. One is the finance-related [*9] and the other has a focus for geometry [*10]. In the probability document, structures of random walk steps "$N^{7/4}$" and in the finance document, the writing of samples "(x_t, x_{t-1})" in probability showed relatively higher feature weight than textual features do, which could contribute to successful classification.

6. Discussion

From the experimental results, our proposed method marks the best classification results among the methods. Result analysis shows the successful examples of our proposed method. The improvement comes from specific categories where the structures of ME can increase classification performance such as Probability and Representation theory in mathoverflow collection.

Aside from proposed method, although BOS and CO methods show high recall, F-score of the methods are much lower than BOW in arxiv collection. This is an interesting finding that just adding ME information by stragiht-forward MS modeling cannot

[*9] http://mathoverflow.net/questions/144860/on-mathematical-aspects-of-the-most-recent-nobel-prize-in-economics-winners-wor

[*10] http://mathoverflow.net/questions/158811/wander-distance-of-self-avoiding-walk-that-backs-out-of-culs-de-sac

Table 9 Significance results between existing and proposed method in mathoverflow results. The numbers in heading are method ID in Table 8. "*" and "**" indicates the significance level at $p < 0.05$ and $p < 0.01$ respectively.

	5			6			7		
Category	1	2	4	1	2	4	1	2	4
1				*			**	**	**
2				*		*	**	**	**
3						*	*	**	**
4		*	*				**	**	
5		*	*				**	**	
6			**	**	*		**	**	
7	**			**	**	*		**	**
8				**	**	*		**	**
9	**			**	*	*		**	**
10								*	
11		*		**		**	**		**
12		**		*	*		**		**
13			**	**			**	**	
14				**		**	**		
15				**	**	*	**	**	**

work reflecting the nature of ME that is highly symbolized.

As to F-score, our improvement of proposed method are rather low by 2 to 3 points from text-based classification. There are several categories without significance between proposed method and BOW method in both mathoverflow and arxiv results. It would imply that most of the part of the classification made by the information come from the text and our method can improve the base with structures of ME. Another discussion is on the F-score itself that is at about 0.7 is not high much One of the reasons could be on the nature that a mathematical document is usually multi-categories and highly linked to other subjects. Further study is needed on the relation between the categorial overlaps and the performance.

7. Conclusion

In this paper, we have addressed the MDC with making use of information from ME. For MDC, we proposed methods utilizing the structures of ME in a tree on supervised classification manner, which aims to capture category specific fragments of ME trees. Experimental results showed that F-measure of the proposed method is the higher than the classifications with the state-of-art ME modeling. The result analysis supports that the improvements are brought from utilizing ME structures.

Since the performance of the proposed method depends on expressivity of tree representations, thus how the markup brings the information about ME, for example, not only the information of identifier, variable but also that of a function and so on. This augmentation would be expected to make the classification and search better, which is the what the MathML contents representation and the semantic annotation aims for. So, one of the future works would a parser development which translates the symbolic layout writing into the form with math semantics.

References

[1] Aizawa, Akiko and Kohlhase, Michael: NTCIR-10 Math Pilot Task Overview, *Proceedings of the 10th NTCIR conference*, pp.654–661, (2013).

[2] Aizawa, Akiko and Kohlhase, Michael and Ounis, Iadh: NTCIR-11 Math-2 Task Overview, *Proceedings of the 11th NTCIR conference*, pp.88–98, (2014).

[3] Barthel, Simon and Tönnies, Sascha and Balke, Wolf-Tilo: Large-Scale Experiments for Mathematical Document Classification, *Proceeding of the 15th International Conference on Asia-Pacific Digital Libraries*, pp.83–92, (2013).

[4] Collins, Michael and Duffy, Nigel: New Ranking Algorithms for Parsing and Tagging : Kernels over Discrete Structures, and the Voted Perceptron, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp.263–270, (2002).

[5] Graf, Peter: Substitution tree indexing, Max-Planck-Institut für Informatik, (1994).

[6] Kamali, Shahab and Tompa, Frank Wm.: A new mathematics retrieval system, *Proceedings of the 19th ACM international conference on Information and knowledge management*, pp.1413–1416, (2010).

[7] Kamali, Shahab and Tompa, Frank Wm.: Retrieving documents with mathematical content, *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pp.353–362, (2013).

[8] Kumar, Pavan P. and Agarwal, Arun and Bhagvati, Chakravarthy: A Structure Based Approach for Mathematical Expression Retrieval, *Proceedings of the 6th International Workshop on Multi-disciplinary Trends in Artificial Intelligence*, pp.23–34, (2012).

[9] Kohlhase, Michael and Sucan, Ioan: A Search Engine for Mathematical Formulae, *Proceedings of the 8th Internationall conference on Artificial Intelligence and Symbolic Computation*, pp.241–253, (2006).

[10] Lin, Xiaoyan and Gao, Liangcai and Hu, Xuan and Tang, Zhi and Xiao, Yingnan and Liu, Xiaozhong: A mathematics retrieval system for formulae in layout presentations, *Proceedings of the 37th international ACM SIGIR conference on Research and development in Information Retrieval*, pp.697–706, (2014).

[11] Lin, Xiaoyan and Gao, Liangcai and Tang, Zhi and Baker, Josef and Sorge, Volker: Mathematical formula identification and performance evaluation in PDF documents, *International Journal on Document Analysis and Recognition*, Vol.17, No.3, pp.239–255, (2014).

[12] Miller, Bruce R. and Youssef, Abdou: Technical aspects of the Digital Library of Mathematical Functions, *Annals of Mathematics and Artificial Intelligence*, Vol.38, No.1, pp.121–136, pp.121–136, (2003).

[13] Miner, Robert and Munavalli, Rajesh: An Approach to Mathematical Search Through Query Formulation and Data Mormalization, *Proceedings of the 6th International conference on Mathematical Knowledge Management*, pp.342–355, (2007).

[14] Moschitti, Alessandro: Efficient convolution kernels for dependency and constituent syntactic trees, *Proceedings of the 17th European Conference on Machine Learning*, pp.318–329, (2006).

[15] Nguyen, Tam T and Chang, Kuiyu and Hui, Siu Cheung: A Math-Aware Search Engine for Math Question Answering System, *Proceedings of the 21st ACM international conference on Information and knowledge management*, pp.724–733, (2012).

[16] Nguyen, Tam T and Hui, Siu Cheung and Chang, Kuiyu: Expert Systems with Applications A lattice-based approach for mathematical search using Formal Concept Analysis, *Expert Systems With Applications*, Vol.39, No.5, pp.5820–5828, (2012).

[17] Řehůřek, Radim and Sojka, Petr: Automated Classication and Categorization of Mathematical Knowledge, *Proceedings of the 7th International Conference on Mathematical Knowledge Management*, pp.543–557, (2008).

[18] Severyn, Aliaksei and Moschitti, Alessandro: Structural Relationships for Large-Scale Learning of Answer Re-ranking Categories and Subject Descriptors, *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pp.741–750, (2012).

[19] Shawe-Taylor, John and Cristianini, Nello: The MCAT Math Retrieval System for NTCIR-10 Math Track, Kernel Methods for Pattern Analysis, Cambridge University Press, (2004).

[20] Topic, Goran and Kristianto, Giovanni Yoko and Nghiem, Minh-Quoc: The MCAT Math Retrieval System for NTCIR-10 Math Track, *Proceedings of the 10th NTCIR Conference*, pp.680–685, (2013).

[21] Watt, Stephen M.: Mathematical Document Classification via Symbol Frequency Analysis, *Proceedings of the workshop Towards a Digital Mathematics Library*, pp.29–40, (2008).

[22] Yan, Xifeng and Han, Jiawei: gSpan: Graph-Based Substructure Pattern Mining, *Proceedings of the IEEE International Conference on Data Mining*, pp.721–724, (2002).

[23] Zanibbi, Richard and Blostein, Dorothea: Recognition and retrieval of mathematical expressions, *International Journal on Document Analysis and Recognition*, Vol.15, No.4, pp.331–357, (2016).

[24] Zanibbi, Richard and Aizawa, Akiko and Kohlhase, Michael and Ounis, Iadh and Topic, Goran and Davila, Kenny: NTCIR-12 MathIR Task Overview, *Proceedings of the 12th NTCIR conference*, pp.299–308, (2016).