PAPER

Using MathML Parallel Markup Corpora for Semantic Enrichment of Mathematical Expressions

Minh-Quoc NGHIEM[†a)], **Giovanni YOKO KRISTIANTO**[††b)], ***Nonmembers***, ***and*** **Akiko AIZAWA**[†††c)], ***Member***

SUMMARY This paper explores the problem of semantic enrichment of mathematical expressions. We formulate this task as the translation of mathematical expressions from presentation markup to content markup. We use MathML, an application of XML, to describe both the structure and content of mathematical notations. We apply a method based on statistical machine translation to extract translation rules automatically. This approach contrasts with previous research, which tends to rely on manually encoded rules. We also introduce segmentation rules used to segment mathematical expressions. Combining segmentation rules and translation rules strengthens the translation system and archives significant improvements over a prior rule-based system.

key words: *semantic enrichment, MathML markup, statistical machine translation*

1. Introduction

1.1 Motivation

The semantic enrichment of mathematical documents is among the most significant areas of discussion related to the digitization of mathematical and scientific content and its applications. The challenge entails associating semantic tags, usually concepts, with mathematical expressions. Encoding the underlying mathematical meaning of an expression confers several benefits: (1) It facilitate more precise information exchange between systems that process mathematical objects; (2) It improves search accuracy by enabling semantic searching of mathematical expressions; (3) It also benefits computer algebra systems, automated reasoning systems, and multi-lingual translation systems.

1.2 Challenges

As with natural language, the semantic enrichment of mathematical expressions is a nontrivial task. Although more rigorous than natural language, mathematical notations are ambiguous, context-dependent, and vary from community to community. The difficulty in inferring semantics from a presentation stems from the many-to-many potential mappings from presentation to semantic [1]. Examples include binomial coefficients, which can be presented in varying notations: $C(n,k)$, ${}_nC_k$, ${}^nC_k$, C_n^k, C_k^n. Moreover, each notation can have other author-dependent meanings aside from the binomial coefficient itself.

This paper introduces an automatic semantic enrichment method for mathematical statements to analyze and disambiguate mathematical terms. We use MathML [1] Presentation Markup to display mathematical expressions and MathML Content Markup to convey mathematical meaning. The semantic enrichment task then becomes the task of generating Content MathML outputs from Presentation MathML expressions.

1.3 Limitations of Prior Work

Prior attempts to address this problem include SnuggleTeX [2] and LaTeXML [3]. These systems use handwritten rule-based methods for disambiguation and translation. Two issues limit these solutions: (1) As handwritten rule-based systems, these systems require mathematical knowledge and human involvement; (2) These systems remain at the experimental stage due to difficulties with processing complex mathematical symbols and due to the wide-ranging nature of mathematical expressions.

1.4 Our Approach

This paper proposes an approach based on Statistical Machine Translation (SMT) [4] techniques. In the proposed framework, the underlying mathematical meaning of an expression is inferred from the probability distribution $p(c|p)$ that a semantic expression c is the translation of presentation expression p. The probability distribution is automatically calculated given parallel markup MathML data which contains both Presentation and Content MathML markup for a single expression. The data used in this study was collected from the Wolfram Functions Site [5] (WFS). We also prepared other parallel markup MathML data by annotating mathematical expressions drawn from 20 papers from the ACL Anthology Reference Corpus [6] (ACL-ARC).

We performed a ten-fold cross validation on mathematical expressions from the six categories of the Wolfram Functions Site. This experiment evaluated the effectiveness of our learning method. We performed another experiment to assess the correlation between systems performance and

Manuscript received October 9, 2012.
Manuscript revised February 23, 2013.

[†]The author is with The Graduate University for Advanced Studies, Tokyo, 101–8430 Japan.
[††]The author is with The University of Tokyo, Tokyo, 113–8654 Japan.
[†††]The author is with the National Institute of Informatics, Tokyo, 101–8430 Japan.
a) E-mail: nqminh@nii.ac.jp
b) E-mail: giovanni@nii.ac.jp
c) E-mail: aizawa@nii.ac.jp

DOI: 10.1587/transinf.E96.D.1707

training set size. We found that increasing the size of the training data boosts systems performance. We compared our method to prior work [2] using a data set collected from ACL-ARC scientific papers. We found that our approach yields improvements in the mathematics semantic enrichment problem, generating fewer errors and outperforming previous work.

1.5 Key Contributions

This paper provides contributions in two main areas:

- First, this research is a first attempt to apply machine translation techniques to the problem of mathematical semantic enrichment. Our experimental results showed that the proposed framework can successfully handle many practical instances of the semantic enrichment that was not possible with conventional rule-based systems.
- Second, the system gains mathematical knowledge (i.e., symbols meanings, structural relationships) automatically during the training process. As long as we have training data available, the system is easily updated and augmented. Since new notations continue to arise, fast, automatic updates are key to the usefulness of any system. SMT-based method only needs to learn parallel corpus to generate a translation engine. In contrast, a rule-based system needs a great deal of knowledge external to the corpus that only mathematical experts can generate. Making such parallel data is easier than making new translation rules, based on our experience.

Additionally, we introduce a metric (the Tree Edit Distance Rate) for evaluating the quality of the tree machine-translated from one markup language to another. Using this metric, we can avoid human judgments of evaluation which are expensive and noisy.

The remainder of this paper is organized as follows. Sections 2 and 3 provide a brief overview of the background and related work on semantic enrichment of mathematical expressions. Section 4 presents our method. Section 5 describes the experimental setup and results. Section 6 concludes the paper and points to avenues for future work.

2. Markup Languages for Mathematical Formulas

A special markup is required to represent mathematical formulas, since mathematical formulas contain both mathematical symbols and structures. Until recently, mathematical formulas have been presented on the Web as images. While this approach requires no markup language to decode, the resulting presentation is hard to process. One way of dealing with mathematical formulas presented in this format is to convert them by optical character recognition into another text-based format. One example is InftyReader [7].

TEX has been widely used to encode mathematical formulas in scientific documents. TEX provides a text syntax for mathematical formulas so that authors can typeset equations in their papers by themselves. A formula is printed as a person would write it by hand, or as a person would typeset the equation. On certain web pages, such as Wikipedia, formulas are displayed in both image and TEX formats.

The best-known open markup format for representing mathematical formulas for the web is MathML [1], a format recommended by the W3C Math Working Group as a standard to represent mathematical expressions. MathML is an XML application for describing mathematical notations and encoding mathematical content within a text format. MathML has two types of encoding: content-based encoding, called Content MathML, which addresses the meaning of formulas; and presentation-based encoding, called Presentation MathML, which addresses the display of formulas. The presentation elements of Presentation MathML are divided into two classes: token elements and layout schemata. Token elements represent identifier names, function names, numbers, and so forth. Layout schemata build expressions from parts. Figure 2 (a) shows the illustration trees for the Presentation and Content Markup of the expression $sin(\frac{\pi}{6}) = \frac{1}{2}$. Other markups are available beyond MathML, including eqn [8], OpenOffice.org Math [9], ASCIIMathML [10], and OpenMath [11]. All these markups can be converted into MathML using freely available tools [12].

We chose to use MathML markup for the following reasons:

- Since its release in 1997, MathML has grown to become a general format that enables mathematics to be served, received, and processed in a wide range of applications.
- MathML can be used to encode both mathematical notation and mathematical content.
- Large collections of formulas are already available in MathML, and access to these collections is relatively easy.

3. Related Work

3.1 Semantic Enrichment for Mathematical Formulas

Few studies have addressed the semantic enrichment problem. In this section, we list some of the work on interpreting the meaning of mathematical expressions.

Grigole et al. [13] proposed an approach to understand mathematical expressions based on the text surrounding the mathematical expressions. The main concept underlying this approach is to use the surrounding text for disambiguation based on word sense disambiguation and lexical similarities. First, a local context C (five nouns preceding a target mathematical expression) is found in each sentence. For each noun, the system identifies a Term Cluster (derived from the OpenMath Content Dictionary). The highest semantic scores obtained are weighted, summed up, and normalized by the length of the considered context. The Term Cluster with the highest similarity score is assigned as the

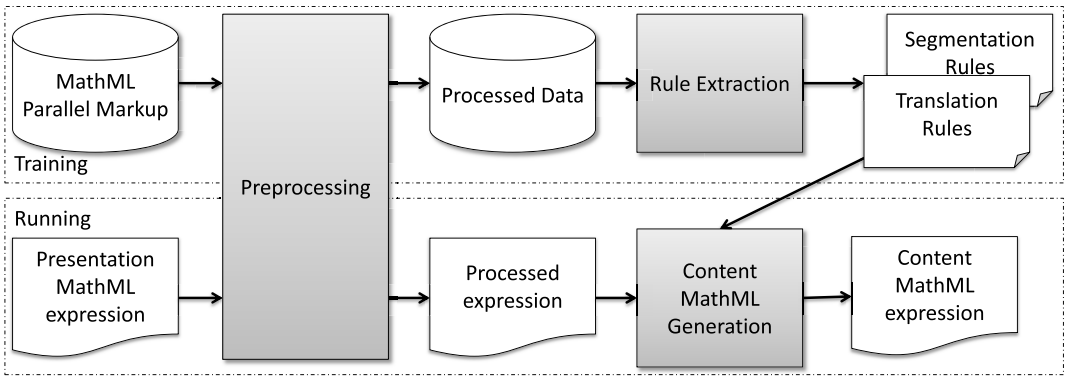

Fig. 1 System framework.

interpretation. When this approach was evaluated for 451 manually annotated mathematical expressions, the best result was an $F_{0.5}$ score of 68.26. To address the meanings of mathematical formulas, Nghiem et al. [14] proposed an approach for extracting names or descriptions of formulas from the natural language text. The most accurate extraction result using data from Wikipedia was 68.33 percent.

Two other current projects address the semantic interpretation of mathematical expressions. The first project is the SnuggleTeX [2], which provides a free and open-source Java library for converting fragments of LaTeX into XML, including Content MathML. The other project is Lamapun [15]. This project investigates semantic enrichment, structural semantics, and ambiguity resolution in mathematical corpora. The project uses LaTeXML [3] to convert from LaTeX to MathML. Unfortunately, no evaluations of these systems have been made to date.

For the generation of Content MathML, SnuggleTeX uses a set of manually encoded transformation rules. The current version supports operators that are the same as ASCIIMathML. For example, it uses the ASCII string "*in*" instead of the symbol "∈". Unlike SnuggleTeX, LaTeXML does not provide a direct way to generate Content MathML from Presentation MathML.

3.2 Application of Statistical Machine Translation

Statistical machine translation (SMT) [4], [16]–[18] is by far the most widely studied machine translation method. SMT uses a very large data set of good translations that is, a corpus of texts already translated into another language. It uses these texts to automatically infer a statistical model of translation. The statistical model is then applied to new texts to derive a translation. Word Alignment-based Semantic Parsing [19] applies MT techniques to learn semantic parsers. The basic idea is to use SMT to learn to translate from natural language to meaning representation language. A word alignment model is used for lexical acquisition, and a syntax-based translation model is used as the parsing model. This study shows SMT can be applied successfully to semantic parsing.

4. Proposed System

4.1 System Overview

We applied the same framework as SMT systems here. We use the parallel markup expressions to automatically infer a statistical model of translation (rules for translation and their probabilities). The statistical model is then applied to new expressions to derive a translation. Figure 1 gives the system framework. The system has two phases, a training phase and a running phase, and consists of three main modules.

- Preprocessing: Processes MathML expressions. It removes error expressions and XML tags that convey no meaning.
- Rule Extraction: Extracts rules for translation, given the training data. We establish two types of rules: segmentation rules and translation rules. Each rule is associated with its probability.
- Content MathML Generation: Generates Content MathML expressions from input Presentation MathML expressions using rules from the Rule Extraction step.

4.2 Preprocessing

In MathML Presentation Markup, certain elements are used for formatting purposes only: the *mtext* and *mspace* tags are used to insert a space between expressions. Some *mtable* tags are used to number the mathematical expressions. A pair of parentheses indicates that the expressions in the parentheses belong together. We removed these elements in specific cases where the structure encodes the same information. Keeping these elements can produce misleading results. Figure 2 (b) illustrates an example of this step.

Expressions with more than 200 nodes in their Content Markup are removed for simplification.

The data contains expressions that convey the same meaning, but their Content MathML are written in different ways. To improve the alignment results, we normalized two expressions having the same content meaning on the Content MathML side. Currently, we implemented these

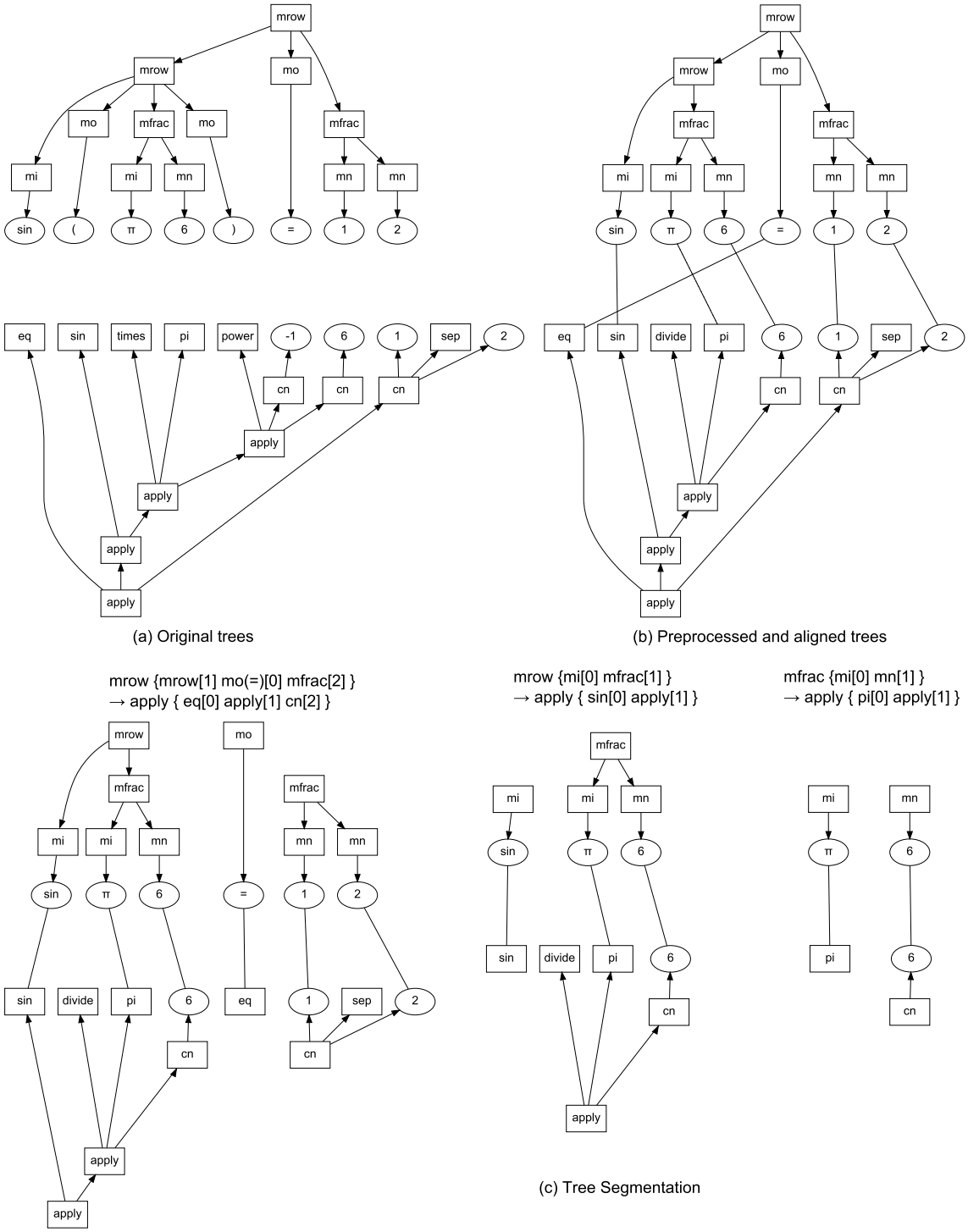

Fig. 2 (a) Original Presentation and Content MathML Markup tree representations (b) preprocessed trees and the alignment between the nodes (c) segmentation process.

three cases in our system: (1) $sqrt(X)$ and $X^{\frac{1}{2}}$, (2) $X - Y$ and $X + (-Y)$, and (3) $\frac{1}{X}$ and X^{-1}.

4.3 Extracting Rules

We extracted two sets of rules: segmentation rules and translation rules. Segmentation rules are used to segment Presentation MathML trees into smaller subtrees. Translation rules are used to translate Presentation MathML trees into Content MathML trees. Segmentation rules and translation rules operate the same as "grammar rules" and "rule table" in SMT systems.

4.3.1 Extracting Segmentation Rules

We proposed segmentation rules to divide a large Presentation MathML tree into smaller subtrees while maintaining alignment with their corresponding Content MathML trees. Long sentences pose a common problem for SMT. System training with long sentence pairs requires more memory and CPU time. The translation quality is also low due to poorly aligned words in long sentence pairs. In our study, 151.2 nodes is the average length of mathematical expressions in the dataset (counting only the leaf nodes). The 30.66 average node is still high, even after removing expressions with more than 200 nodes in their Content Markup. Long mathematical expressions must be segmented into shorter ones. Note that segmenting MathML expressions is easier than segmenting natural language sentences since the structural information is explicitly encoded using XML.

For a given mathematical expression pair (p, c), we have $p_1, p_2, \ldots, p_n$ as subtrees of p and $c_1, c_2, \ldots, c_m$ as subtrees of c. A segmentation of (p, c) is defined as a sequence of subtree pairs $(p_{s_1}, c_1), (p_{s_2}, c_2), \ldots, (p_{s_m}, c_m)$, where p_{s_1}, $p_{s_2}, \ldots, p_{s_m}$ are corresponding subtrees of $c_1, c_2, \ldots, c_m$.

To achieve segmentation, we use GIZA++ [20] to obtain alignment between the leaf nodes of Presentation and Content MathML trees. Figure 2 (b) shows an example of this alignment. We extract Segmentation Rules based on this alignment. For each Content MathML subtree c_i, the corresponding Presentation MathML subtree p_{s_i} is the subtree satisfying the following condition:

$$p_{s_i} = \arg\max_j P(p_j|c_i, a) \tag{1}$$

$P(p_j|c_i, a)$ is calculated by obtaining the ratio of number of alignments between p_j and c_i to the total of alignment in a, where variable a represents the alignments between p and c.

$$P(p_j|c_i, a) = \frac{count[a(p_j, c_i)]}{|a|} \tag{2}$$

We apply the following constraint: distinct Presentation subtrees cannot be aligned with the same Content subtree. The only exception is the case of operators. Many identical operators subtrees in a Presentation subtree can be aligned with one Content subtree. We make this allowance because the Content function can have more than two arguments, while the Presentation operator permits only two. A segmentation that does not satisfy this constraint is invalid. A segmentation rule is created each time we segment the tree. Each segmentation rule is associated with a probability which represents how likely it is that the right-hand side of the rule will happen given the left-hand side.

Figure 2 (c) shows an example of segmentation process and extracted segmentation rules. Table 1 gives examples of segmentation rules. In the table, the numbers, such as [1], represent corresponding Presentation and Content markup subtrees.

Table 1 Examples of segmentation rules extracted from Wolfram Functions Site dataset.

Segmentation Rule	Probability
mrow { mrow[1] mo(=)[0] msup[2] } → apply { eq[0] apply[1] apply[2] }	1
mrow { mrow[1] mo(/;)[0] mrow[2] } → apply { ci(Condition)[0] apply[1] apply[2] }	0.9998
mrow { mrow[1] mo(=)[0] mrow[2] } → apply { eq[0] apply[1] apply[2] }	0.9946
mrow { mrow[1] mo(∝)[0] mrow[2] } → apply { ci(Proportional)[0] apply[1] apply[2] }	0.9511
mrow { msup[1] mo(.)[0] mrow[2] } → apply { times[0] apply[1] apply[2] }	0.8582

Table 2 Examples of translation rules extracted from Wolfram Functions Site dataset.

Translation Rule	Probability
< mo > . < /mo > → < times/ >	1
< mo > ∈ < /mo > → < in/ >	1
< mi > m < /mi > → < ci > m < /ci >	1
< mo > /; < /mo > → < ci > Condition < /ci >	0.9998
< mo > = < /mo > → < eq/ >	0.9993
< mi > n < /mi > → < ci > n < /ci >	0.9941
< mo > - < /mo > → < minus/ >	0.9431
< mo > - < /mo > → < plus/ >	0.0566
< mo > + < /mo > → < plus/ >	0.9995

4.3.2 Extracting Translation Rules

If we cannot segment the subtree or if the segmentation is invalid, we extract a translation rule. Translation rules are used to translate a Presentation tree directly into a Content tree. Each translation rule is also associated with its frequency of occurrence throughout the training process. Training halts when no expressions can be segmented. Algorithm 1 gives the pseudo code for extracting the rules. Function "UpdateProbability" uses Eq. (2) to calculate the probability of each rule. Function "GetTranslationRule" and "GetSegmentationRule" extract the appropriate rules from the traning sample. Function "ExtractRule" calls ifself recursively until the subtree cannot be segmented anymore. Table 2 shows examples of translation rules. Note that the rule <mo>-</mo> → <plus/> is a legal translation rule but its probability is low. The rule is extracted from those expressions which contain addition of 3 or more terms, i.e. $X - Y + Z$ (plus between X and $-Y$ and Z), these expressions were not normalized in the preprocessing step. Alignment errors or segmentation errors can also lead to wrong rule extraction.

4.4 Content MathML Generation

We apply segmentation rules and translation rules for the translation at this step. Given a Presentation MathML tree, the system will generate a corresponding Content MathML tree. We use a greedy translation method here to reduce translation time. If more than two rules can be applied to translate a tree, the rule with higher probability is chosen.

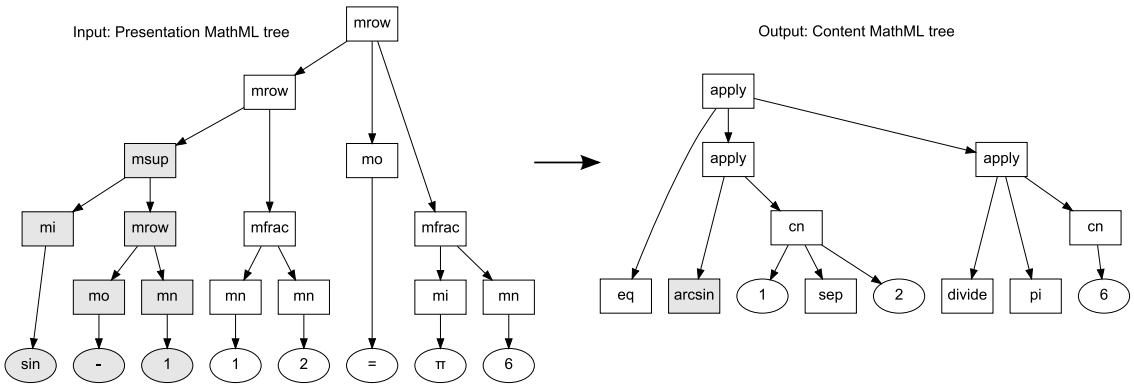

Fig. 3 Translation of $sin^{-1}\frac{1}{2} = \frac{\pi}{6}$. When the term <mi>sin</mi> is accompanied by <mrow><mo>-</mo><mn>1</mn></mrow>, the system can correctly translated it to <arcsin/>.

Algorithm 1 Extract Translation Rules and Segmentation Rules

Input: set of training MathML files parallel markup M
Output: list of segmentation rules SR
list of translation rules TR
function EXTRACTRULES(M)
$SR \leftarrow \emptyset$
$TR \leftarrow \emptyset$
A = ALIGN(M) ▷ alignments of nodes (output of GIZA++)
for all $m \in M$ **do**
EXTRACTRULE(m,A,SR,TR)
end forreturn SR, TR
end function

function EXTRACTRULE(m,A,SR,TR)
tr = GETTRANSLATIONRULE(m) ▷ Extract the translation rule
if TR contains tr **then**
UPDATEPROBABILITY(TR)
else
$TR \leftarrow TR \cup \{tr\}$
end if
sr = GETSEGMENTATIONRULE(m) ▷ Extract the segmentation rule
if SR contains sr **then**
UPDATEPROBABILITY(SR)
else
$SR \leftarrow SR \cup \{sr\}$
end if
let subTrees[1 .. N] be subtrees of m
for $i = 1 \rightarrow N$ **do**
EXTRACTRULE($subTrees[i]$,A,SR,TR)
▷ Extract rules of each subtree
end for
end function

The translation process is as follows: First, we apply the same preprocess module on the Presentation MathML tree. The difference here is that we remove only non-semantic elements. Second, if the processed tree can be translated using translation rules, then apply the rule for translation. If not, the segmentation rule is applied to segment the tree into subtrees. If no rule can be applied, return a translation error. Third, the translation rules are applied to translate the Presentation MathML subtrees into Content MathML subtrees. Finally, the translated Content MathML subtrees are grouped to form the complete Content MathML tree.

Algorithm 2 Translate Presentation to Content MathML tree

Input: Presentation MathML tree preTree
segmentation rules SR
translation rules TR
Output: Content MathML tree contentTree
function TRANSLATE($preTree$)
rule1 ← GETBESTRULE(preTree, TR)
if $rule1 \neq null$ **then**
return APPLY(tRule, preTree)
end if
rule2 ← GETBESTRULE(preTree, SR)
if $rule2 \neq null$ **then**
let pSub[1 .. N] be subtrees of preTree
let cSub[1 .. N] be new contentTree
for $i = 1 \rightarrow N$ **do**
$cSub[i]$ = TRANSLATE((pSub[i]))
end for
return REBUILDTREE(cSub, sRule)
▷ combines cSub based on the segmentation rule
else
return < $cerror$/ >
end if
end function

Algorithm 2 gives the translation algorithm. The "GetBestRule" function searches for the rule with highest probability in the rule list. The "Apply" function applies a translation rule to a Presentation MathML tree and returns the translated Content MathML tree. The "RebuildTree" function combines the translated subtrees into a complete tree based on the alignment indexes in the segmentation rule. In some cases, we were unable to apply any of the segmentation or translation rules, generally due to unseen data. For those cases, the system ignored the root of the subtree and translated its children. This would generate errors at the root of the subtree but improve overall performance. We also applied some heuristic translations to translate numbers and identifiers in the *mn* and *mi* tags.

Using the proposed approach, the system is capable of handling ambiguous cases. Figure 3 shows an disambiguation example. Normally the term <mi>sin</mi> is translated to <sin/> but when it is accompanied by <mrow><mo>-</mo><mn>1</mn></mrow>, the system

can correctly translated it to <arcsin/>.

5. Experimental Results and Discussions

5.1 Evaluation Setup

We used two datasets for the experiments: (1) The first dataset is WFS and contains mathematical expressions collected from the Wolfram Functions site [5], a site created as a resource for educational, mathematical, and scientific communities. The site contains the world's most encyclopedic collection of information on mathematical functions. All formulas on this site are available in both Presentation MathML and Content MathML format. In our experiments, we chose six categories that contains 205,653 mathematical expressions in total. (2) The second dataset is ACL Anthology Reference Corpus [6] (ACL-ARC) which contains mathematical expressions extracted from scientific papers in the area of Computational Linguistics and Language Technology. This corpus is also a target corpus of the mathematical formula recognition task in The ACL 2012 Contributed Task [21]. Currently, we use mathematical expressions drawn from 20 papers to investigate the cross-domain applicability of the proposed method. We have manually annotated all mathematical expressions in these papers with both Presentation Markup and Content Markup. The total number of mathematical expressions in the data set is 2,065. Table 3 gives various statistics for these datasets.

We used the default parameter setting of GIZA++ to obtain the alignments between Presentation MathML terms and Content MathML terms.

5.2 Evaluation Methodology

Training and testing were performed using ten-fold cross-validation. For each category, we partitioned the original corpus into ten subsets. Of the ten subsets, we retained a single subset as validation data for testing the model, using the remaining subsets as training data. The cross-validation process was repeated ten times, with each of the ten subsets used exactly once as validation data. The ten results from the folds then averaged to produce a single estimate. In both datasets, we used formula-wise partition.

Given a Presentation MathML expression e, let A is the correct Content MathML tree and B is the output tree of the automatic translation. We evaluate the correctness of tree B by comparing it directly to tree A. In the experiments, we extended the conventional definition of Translation Edit Rate and applied a specific metric that combines the following:

- Tree Edit Distance [22]: The tree edit distance is the minimal cost of transforming A into B using edit operations. Three types of edit operations are possible: substituting, inserting, or deleting a node.
- Translation Edit Rate [23]: The translation edit rate is an error metric for machine translation that measures the number of edits required to change a system output into one of the references.

We call the new metric the Tree Edit Distance Rate (TEDR). TEDR is defined as the ratio of (1) the minimal cost of transforming tree A into another tree B using edit operations and (2) the maximum number of nodes of A and B. It can be computed using Eq. (3).

$$TEDR(A \rightarrow B) = \frac{TED(A,B)}{max\{|A|,|B|\}} \quad (3)$$

For example, Fig. 4 depicts an output tree (A) and a reference (B). Compared to the reference tree, we must substitute 1 node, insert 3 nodes, and delete 0 node in the output tree, so that $TED(A,B) = 4$, while the maximum number of nodes of the two trees is 8. Therefore, $TEDR(A \rightarrow B) = \frac{4}{8} = 0.5$. $TEDR = 0$ is optimal for this metric.

Besides TEDR, we used Perfect Translation Rate (PTR). PTR is simply the percentage of perfectly translated expressions.

5.3 Experimental Results

First, we investigated the coverage of segmentation and translation rules which were automatically extracted from the training data. We used the data from the Elementary Functions category, the largest category. Segmentation and translation rules are effective in 98.69% of translation cases. The rest 1.31% is where we cannot apply any segmentation nor translation rule, which will generate *cerror* node. Translation rules are used twice as often as segmentation rules. Translation rules contribute 65.62% of the translation, while segmentation rules contribute about 33.07%. The accuracy of these rules is 99.13% and 98.3%, respectively.

Table 3 Data statistics. The first six categories were collected from the Wolfram Functions site. The last was extracted from 20 ACL papers.

Category	No. of math expressions
Bessel-TypeFunctions	1,960
Constants	709
ElementaryFunctions	30,220
GammaBetaErf	2,895
IntegerFunctions	1,612
Polynomials	1,489
ACL-ARC	2,065

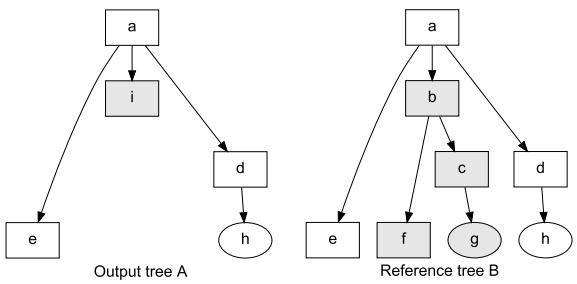

Fig. 4 Example of an output tree (A) and a reference (B). $TEDR(A \rightarrow B) = 0.5$.

Table 4 Results for each category of the Wolfram Functions Site data. The second and third columns show the average number of segmentation rules and translation rules extracted on each fold, respectively. The two last columns show TEDR and PTR scores.

Category	Avg. No. of FR	Avg. No. of TR	TEDR	PTR
Bessel-TypeFunctions	447	9,432	42.31	19.24
Constants	258	1,116	42.35	18.67
ElementaryFunctions	937	12,286	**8.00**	**67.48**
GammaBetaErf	658	8,594	49.30	15.9
IntegerFunctions	431	2,667	41.03	23.2
Polynomials	457	4,464	45.73	13.04

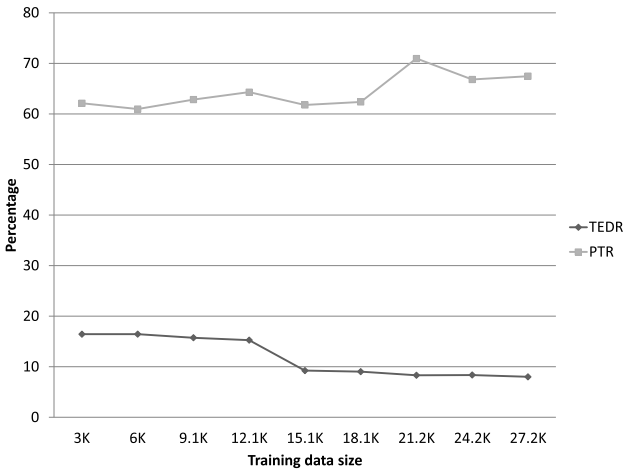

Fig. 5 Correlation between TEDR and PTR scores and training set size.

(This value is calculated by the ratio of the correct rules applied to the total rules applied.) The results show that the coverage of segmentation and translation rules is high and selected rules are mostly correct.

We then investigated the translation quality of the system with different categories. For the WFS dataset, our experimental results in Table 4 showed our approach gave good results: an 8% TEDR score with a large training data set ("Elementary Functions" category). For smaller data sets (fewer than 3,000 training samples), the results vary from 41% to 49% TEDR.

Third, we set up an experiment using mathematical expressions in the Elementary Functions category. This experiment investigated the correlation between translation quality and the size of the training data set. We used a fixed set of 3,022 expressions for testing. The size of the training data varied from 3,022 to 27,198 expressions. Figure 5 shows the correlation between translation quality and training set size. The larger the training data, the better the results. The evaluation also shows that 15,000 training examples are insufficient to achieve stable results (fewer than 10% TEDR).

Finally, we set up an experiment to compare our system to SnuggleTeX using ACL-ARC dataset[†]. We set up two systems, SMT-1 used ACL-ARC data for training and testing while SMT-2 used WFS data for training and ACL-ARC data for testing. Table 5 shows the TEDR and PTR scores of our systems compared to SnuggleTeX. SMT-2

Table 5 Results for ACL-ARC data. SMT-1 used ACL-ARC data, ten-fold cross-validation. SMT-2 used the rules extracted from Wolfram Functions Site Data.

	TEDR	PTR
SMT-1	**58.63**	**47.12**
SMT-2	63.65	35.17
SnuggleTeX	91.32	30.77

system had a 27.67% lower TEDR score and a 4.4% higher PTR score compared to SnuggleTeX. For this cross-domain setting, SMT-based method is advantageous, and even more when the datasets belong to the same domain. SMT-1 system had a 32.69% lower TEDR score and a 16.35% higher PTR score compared to SnuggleTeX, while running times of both systems were more or less equivalent. However, our systems needed to learn the rules from the training data in advance. The result of SMT-1 is higher than the result of SMT-2 because this system took advantage of the manually annotated training data of papers from the ACL archive.

6. Conclusions and Future Work

This paper discussed the problems posed by the semantic enrichment of mathematical expressions. Our results show an approach based on statistical machine translation for translating Presentation MathML expressions into Content MathML expressions represents a significant improvement over a prior rule-based system. Mathematical notations are context-dependent, so to generate the correct semantic output, we must consider not just the surrounding expressions but also the document containing the notations. In the scope of this paper, we considered only the first kind of context information. This being merely a first attempt at translation from Presentation to Content MathML using machine learning methods, room for improvement certainly remains. Potential improvements include the following:

- Expanding training data so the system can cover more mathematical notations from different categories.
- Incorporating the information implicit in surrounding mathematical expressions; for example, definitions or other mathematical expressions.
- Improving alignment accuracy. Alignment errors can generate errors in the subsequent steps of the translation, such as rule extraction.

Our approach, which combines automatic extraction of segmentation rules and translation rules, shows promise. Experimental results confirm it should aid in the automatic understanding of mathematical expressions. This, however, is merely a first step. Many important issues remain for future study. Our system currently handles a limited range of mathematical notations. Future efforts should seek to expand the systems capacity to handle all mathematical notations.

[†]SnuggleTeX cannot be used with the WFS dataset because the WFS dataset contains a large number of Unicode symbols while SnuggleTeX provides very limited support.

Acknowledgment

This research has been partially supported by Japan Society for the Promotion of Science (JSPS) Grant In-Aid for Scientific Research (B) under grant 24300062.

References

[1] R. Ausbrooks, et al., "Mathematical markup language (MathML) version 3.0," W3C Recommendation, World Wide Web Consortium, 2010.
[2] D. McKain, "SnuggleTeX version 1.2.2."
[3] B. Miller, "LaTeXML a LaTeX to XML converter." http://dlmf.nist.gov/LaTeXML/
[4] P.F. Brown, J. Cocke, S.A.D. Pietra, V.J.D. Pietra, F. Jelinek, J.D. Lafferty, R.L. Mercer, and P.S. Roossin, "A statistical approach to machine translation," Computational Linguistics, vol.16, no.2, pp.79–85, 1990.
[5] "The Wolfram Functions Site." http://functions.wolfram.com/
[6] "The archives of the association for computational linguistics." http://acl-arc.comp.nus.edu.sg/
[7] M. Suzuki, T. Kanahori, N. Ohtake, and K. Yamaguchi, "An integrated OCR software for mathematical documents and its output with accessibility," Computers Helping People with Special Needs, Lect. Notes Comput. Sci., vol.3118, pp.648–655, 2004.
[8] B.W. Kernighan and L.L. Cherry, "A system for typesetting mathematics," Commun. ACM, vol.18, no.3, pp.151–157, 1975.
[9] "OpenOffice.org Math." http://www.openoffice.org/product/math.html
[10] "ASCII MathML." http://www1.chapman.edu/jipsen/mathml/asciimath.html
[11] "OpenMath." http://www.openmath.org/
[12] H. Stamerjohanns, D. Ginev, C. David, D. Misev, V. Zamdzhiev, and M. Kohlhase, "MathML-aware article conversion from LaTeX," DML 2009: Proc. 2nd Workshop, pp.109–120, 2009.
[13] M. Grigore, M. Wolska, and M. Kohlhase, "Towards context-based disambiguation of mathematical expressions," Joint Conf. ASCM 2009 and MACIS 2009: Asian Symposium on Computer Mathematics and Mathematical Aspects of Computer and Information Sciences, pp.262–271, 2009.
[14] M.Q. Nghiem, K. Yokoi, Y. Matsubayashi, and A. Aizawa, "Mining coreference relations between formulas and text using Wikipedia," Proc. Second Workshop on NLP Challenges in the Information Explosion Era (NLPIX 2010), pp.69–74, 2010.
[15] D. Ginev, C. Jucovschi, S. Anca, M. Grigore, C. David, and M. Kohlhase, "An architecture for linguistic and semantic analysis on the arXMLiv corpus," Applications of Semantic Technologies, pp.3162–3176, 2009.
[16] P.F. Brown, V.J.D. Pietra, S.A.D. Pietra, and R.L. Mercer, "The mathematics of statistical machine translation: Parameter estimation," Computational Linguistics, vol.19, no.2, pp.263–312, 1993.
[17] D. Chiang, "A hierarchical phrase-based model for statistical machine translation," Proc. 43rd Annual Meeting on Association for Computational Linguistics, pp.263–270, 2005.
[18] K. Yamada and K. Knight, "A syntax-based statistical translation model," Proc. 39th Annual Meeting of the Association for Computational, pp.523–530, 2001.
[19] Y.W. Wong and R. Mooney, "Learning for semantic parsing with statistical machine translation," Proc. 2006 Human Language Technology Conference - North American Chapter of the Association for Computational, pp.439–446, 2006.
[20] F.J. Och and H. Ney, "A systematic comparison of various statistical alignment models," Computational Linguistics, vol.29, no.1, pp.19–51, 2003.
[21] U. Schafer, J. Read, and S. Oepen, "Towards an ACL anthology corpus with logical document structure. An overview of the ACL 2012 contributed task," SIAM J. Comput., 2012.
[22] K. Zhang and D. Shasha, "Simple fast algorithms for the editing distance between trees and related problems," SIAM J. Comput., vol.18, no.6, pp.1245–1262, 1989.
[23] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, "A study of translation edit rate with targeted human annotation," Proc. Association for Machine Translation in the Americas, pp.223–231, 2006.

Minh-Quoc Nghiem obtained his BS in computer science from the Ho Chi Minh City University of Science, Vietnam, in 2006. He is currently a PhD candidate at The Graduate University for Advanced Studies (Sokendai), Japan. His research interests include machine learning, natural language processing, and information retrieval.

Giovanni Yoko Kristianto received his S.T. (Bachelor of Engineering) degree in electrical engineering from Gadjah Mada University, Yogyakarta, Indonesia in 2009. He is currently a masters degree candidate in the Department of Computer Science, Graduate School of Information Science and Technology, The University of Tokyo, Japan. His research interests include natural language processing, information retrieval, and software engineering.

Akiko Aizawa graduated from the Department of Electronics at the University of Tokyo in 1985 and completed her doctoral studies in electrical engineering in 1990. She was a visiting researcher at the University of Illinois at Urbana-Champaign from 1990 to 1992. She is currently Professor at the National Institute of Informatics and at the Graduate School of the University of Tokyo and Visiting Professor at The Graduate University for Advanced Studies (Sokendai). Her research interests include statistical text processing, linguistic resources construction, and corpus-based knowledge acquisition.