Mathematical Biosciences and Engineering

MBE, 19(6): 5428–5445.
DOI: 10.3934/mbe.2022255
Received: 16 November 2021
Revised: 04 March 2022
Accepted: 16 March 2022
Published: 25 March 2022

http://www.aimspress.com/journal/MBE

Research article

Error correction of semantic mathematical expressions based on bayesian algorithm

Xue Wang[1,2], Fang Yang[1,2,*], Hongyuan Liu[1,2] and Qingxuan Shi[1,2]

[1] School of Cyber Security and Computer, Hebei University, Baoding 071002, China
[2] Institute of Intelligent Image and Document Information Processing, Hebei University, Baoding 071002, China

* **Correspondence:** Email: yangfang@hbu.edu.cn.

Abstract: The semantic information of mathematical expressions plays an important role in information retrieval and similarity calculation. However, a large number of presentational expressions in the presentation MathML format contained in electronic scientific documents do not reflect semantic information. It is a shortcut to extract semantic information using the rule mapping method to convert presentational expressions in presentation MathML format into semantic expressions in the content MathML format. However, the conversion result is prone to semantic errors because the expressions in the two formats do not have exact correspondences in grammatical structures and markups. In this study, a Bayesian error correction algorithm is proposed to correct the semantic errors in the conversion results of mathematical expressions based on the rule mapping method. In this study, the expressions in presentation MathML and content MathML in the NTCIR data set are used as the training set to optimize the parameters of the Bayesian model. The expressions in presentation MathML in the documents collected by the laboratory from the CNKI website are used as the test set to test the error correction results. The experimental results show that the average F_1 value is 0.239 with the rule mapping method, and the average F_1 value is 0.881 with the Bayesian error correction method, with the average error correction rate is 0.853.

Keywords: error correction; mathematical expressions; Bayesian algorithm; presentation MathML; content MathML

1. Introduction

In the Information Era, the exchange and sharing of scientific information through electronic documents has been increasingly used, of which the Chinese scientific documents contain a large number of mathematical expressions. Therefore, it is of great significance to research the representation of mathematical expressions for the retrieval of scientific documents [1,2]. Meanwhile, semantic expressions with semantic information play a key role in related research [3–5].

Two kinds of markup to describe mathematical expressions are provided by MathML: presentation markup and content markup [6,7]. Mathematical expressions encoded with presentation markup are referred to as presentational expressions (presentation MathML) which focus on the layout of expressions and can preserve the prototypes of operators and operands, but do not contain semantic information. Mathematical expressions coded with content markup are referred to as semantic expressions (content MathML) which focus on the semantic information for calculation and processing, and contain the internal meaning of the expressions.

The experimental result of Michal et al. [8,9] showed that their search system performs best using content MathML queries. The explanation they given is that with content MathML there is smaller degree of ambiguity than with presentation MathML. These studies use English electronic scientific documents which are included in the NTCIR dataset [10] as datasets for experiments. The English scientific documents in the NTCIR dataset were sorted out by professionals. Each mathematical expression of these documents contains well-formed presentation MathML and content MathML.

However, some Chinese scientific documents contain only presentational expressions (presentation MathML). For example, the Chinese scientific documents collected by the laboratory from natural science journals on the CNKI (China National Knowledge Infrastructure) [11] website. Therefore, we are committed to automatically generate valid content MathML from presentation MathML in the CNKI dataset. The valid content MathML means that it has the correct indentation format and symbol order. In particular, the valid content MathML means that it can represent a complete mathematical expression and be displayed accurately on a webpage. In addition, the valid content MathML can facilitate in-depth research on expression similarity calculation and retrieval. Thus, it has become an urgent problem to obtain content MathML based on presentation MathML.

2. Related works

The problem of format conversion of mathematical expressions has been studied for a long history.

Early on, Zhang et al. [12] used the principles of linked list and stacking and combined with the priority of operators to gradually replace the DOM tree nodes of semantic expressions, so that the interconversion between Content format and Infix format was realized. Hussain et al. [13] used abstract syntax tree to extract the structural information in LaTeX expressions, and generate XML structured mathematical expression. Zhu et al. [14] established a MathML to LaTeX conversion model by analyzing and studying the MathML structure and content information, and realized the conversion from MathML to LaTeX expressions. Su et al. [15] proposed a notation selection strategy to convert from Content MathML to Presentation MathML while using four conversion methods: element-to-element, element-to-text, attribute-to-attribute and structure-to-structure.

Schubotz et al. [16] presented a new approach that combines textual features with the converts to improve the outcome of mathematical format conversions and compared several LaTeX to MathML

converters and they found that many converters simply do not support the conversion from presentation to Content format. Due to the particularity of the markups and internal structure in the expressions in the MathML format, as well as the lack of semantic information in presentational expressions, there is a little research on the conversion between the Presentation format and the Content format. Cai et al. [17] proposed a method for determining ambiguous content in expressions using type prediction. During the expressions conversion process, all symbols in the expression are first identified using a lexical analyzer, and then the type system converts the expression based on the obtained expression information. Doush et al. [18] used a kind of RDFa (Resource Description Framework in attributes) annotation to add the corresponding semantic information framework to expression conversion process. In this way, the expression is converted from presentation MathML to content MathML. Nghiem et al. [19] proposed a system that applied segmentation rules and translation rules to generate the corresponding content MathML tree from the given presentation MathML tree. Toloaca and Kohlhase [20] proposed the MathSemantifier system that converts a meaning tree to content MathML and displays the content MathML trees in the frontend. Using the system, they can convert presentation MathML to content MathML.

Presentational expressions have a strong two-dimensional structure, and pay more attention to the display of mathematical expressions. The complexity and ambiguity of operator notations in mathematical expressions, as Greiner-Petter et al. [21] pointed, have important affect for mathematical expressions format conversion. Grigore et al. [22] presented a preliminary study on disambiguation of symbolic expressions in mathematical documents. Their approach was based on the use of contextual information which is contained in the natural language surrounding the target mathematical expression. Then, disambiguation would be completed by computing a semantic similarity between the words from the lexical context of the given expression and a set of terms from term clusters based on OpenMath. Semantic expressions benefit from more markups advantages and pay more attention to the semantic information of mathematical expressions. For example, the order in which operators and operands of the semantic expression $(a+b)*(c+d)$ appears as $*,+,a,b,+,c,d$. With the rule mapping method, the two-dimensional structure of multiplication cannot be correctly processed, which gives a wrong expression. The order of operators and operands in the wrong expression is $+,a,b,*,+,c,d$, so the expression encoded in content MathML format cannot be displayed accurately on the web page. To obtain an accurate expression, it is necessary to perform error correction on the semantic expression.

Error correction techniques have been widely used in the fields of grammar correction in natural languages [23,24], spelling error correction [25], and text input and recognition [26]. Inspired by the error detection and correction model [27] for input text, this study uses a Bayesian algorithm [28] to correct errors in expressions, aided by edit distance algorithm [29,30], in the process of expression error correction by applying Bayesian error correction algorithm to error correct the wrong expressions.

Bayesian algorithm is a common probability model [31–33] which plays a significant role in the research and development of spam filtering, content recommendation, and spelling error detection, etc. [34–40]. It has the advantages of simple theory, clear algorithm logic, easy implementation, and fast training speed. The Bayesian algorithm is applied to the error correction of mathematical expressions, which solves the problem of low precision of expression format conversion and improves the accuracy of error correction.

The Bayesian algorithm is used in this study to correct errors in mathematical expressions obtained by the rule mapping method. The problem of incorrect display of converted expressions is solved, and expressions with semantic information are obtained. This study provides a benchmark for similarity calculation and expression retrieval system research, as well as ideas and references for the

field of error detection and correction.

3. Error correction of semantic mathematical expressions

The overall process of error correction of semantic mathematical expressions based on the Bayesian algorithm consists of two parts. The overall flow chart is shown in Figure 1.

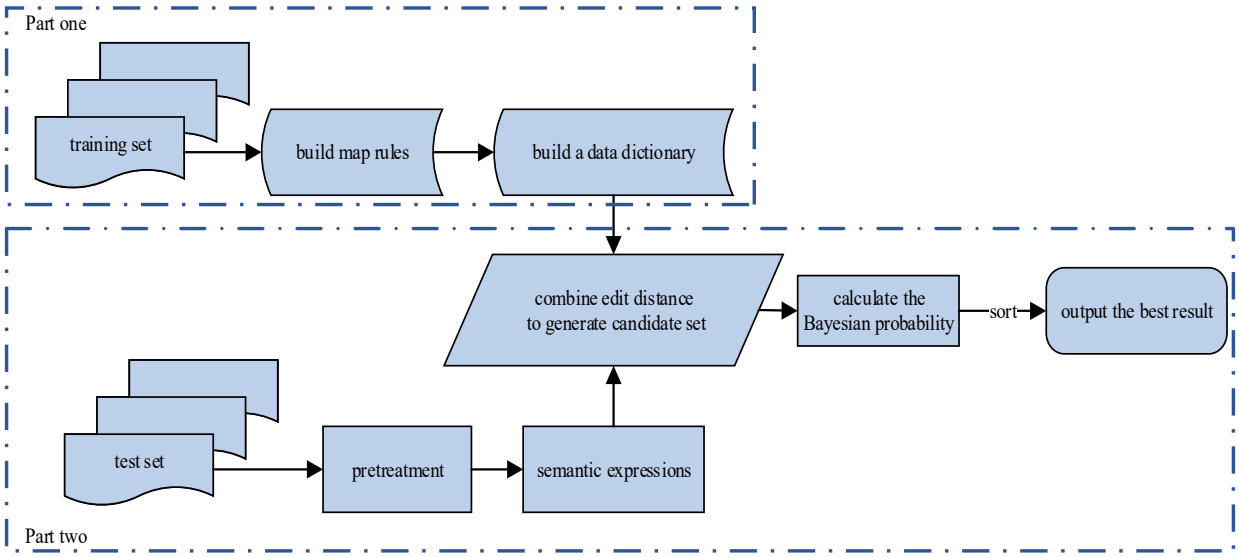

Figure 1. Overall flow chart.

Part one: Presentation MathML and content MathML are extracted from the documents in the training set. According to the content information and statistical results, mapping rules are constructed and a data dictionary is established.

Part two: The expressions in presentation MathML format in the test set are reformatted for correct indentation. The method of rule mapping converts presentational expressions in presentation MathML format into semantic expressions in content MathML format and then we correct those semantic expressions that cannot be presented correctly in web pages. In the error correction process, candidate sets are generated according to the data dictionary and the edit distance algorithm. The Bayesian probability of each expression in candidate sets is calculated after that. The expression with the highest probability is selected as the final result of the error correction expression.

3.1. Mapping rules for expressions markups

Both presentational expressions and semantic expressions contain a large number of markups. Through a large amount of statistical information, it is known that a content markup may correspond several presentation markups. Therefore, creating a set of mapping rules for expressions markups is a key step to obtain semantic expressions and an important prerequisite for subsequent error correction of expressions.

The training set is used to build the mapping rules. In the data set of this study, there are three encoding formats for expressions in English electronic scientific documents: presentation MathML, content MathML and LaTeX expressions. First, the required presentation MathML expressions and

content MathML expressions are extracted from the documents in the training set. Then mapping rules are constructed based on the contents of the expressions' markups and the tree encoding form, and are presented in terms of both content and structure. The specific algorithm is shown in Algorithm 1.

Algorithm 1. Expressions' extraction algorithm.

INPUT: scientific documents: Document
OUTPUT: presentational expression: Pformula; semantic expression: Cformula

```
1 Document ← DirectoryInfo (Path)
2  for <math> in Document // traversing scientific documents
3   <math> ← HtmlParser // resolve mathematical expressions
4   if <math>! = null
5     MathML ← match (@" (<math>\s [^>] ([\s\S]) *?) (</math>)") // extract the expression
6     Pformula ← match (@" (<mrow>\s [^>] ([\s\S]) *?) (</mrow>)", MathML) // Pformula
7     Cformula ← match (@" (<apply>\s [^>] ([\s\S]) *?) (</apply>)", MathML) // Cformula
8   end if
9  end for
10 return Pformula,Cformula
11 END
```

Content: Since both types of expressions have a large of markups, the exact correspondence of the markups is undoubtedly an important part of the process when converting from one type to the other when constructing mapping rules. Table 1 shows the example of expression $a+b$.

Table 1. Two formats of $a+b$.

NO.	presentation MathML	content MathML
1	<math>	<math>
2	<mrow>	<apply>
3	<mi>a</mi>	<plus/>
4	<mo>+</mo>	<ci>a</ci>
5	<mi>b</mi>	<ci>b</ci>
6	</mrow>	</apply>
7	</math>	</math>

The start tag $and its corresponding end tag$ in the first and last lines indicate the expression is expressed in mathematical markup language (MathML). The <mrow> tag in the second line is used to group the subexpressions of an expression. An expression is usually composed of one or more subexpressions. The <apply> tag acts as an encapsulates in the expressions and is the most basic element in semantic markups. In general, the <mrow> tag corresponds to the <apply> tag. The <mi> tag in the fifth line displays the symbolic constants in the expression and corresponds to the <ci> tag in the semantic expression.

In addition, operators in presentation MathML expressions are usually contained by a pair of

<mo></mo>. However, operators in content MathML expressions usually have their own specific tag representations. For example, the markups corresponding to $+$, $-$, $*$, and $\div$ are <plus/>, <minus/>, <times/>, and <divide/>, respectively.

Structure: Table 1 shows that the presentation MathML expressions are typeset in the order in which the operators and operands appear in the expression, while the content MathML expressions are different. The content MathML expressions obtained according to the content of presentation MathML expressions and combined with the mapping rules differ from the accurate content MathML expressions in part of the structure. Therefore, to address the impact of this problem in the experiment, this study constructs a data dictionary through statistics and data analysis and combines it with the Bayesian algorithm to correct wrong expressions and achieve expected results. Table 2 shows the correspondence between some commonly used symbols.

Table 2. Commonly used markups.

symbol	presentation MathML	content MathML	interpretation	example
	<mn>	<cn>	numeral	
	<mi>	<ci>	operator	
+	<mo>+</mo>	<plus/>	addition	$x+y$
-	<mo>-</mo>	<minus />	subtraction	$x-y$
=	<mo>=</mo>	<eq/>	equal	$x=y$
<	<mo><</mo>	<lt/>	less than	$x<y$
∞	<mi>∞</mi>	<infinity/>	infinity	$+\infty$
sin	<mi>sin</mi>	<sin/>	sine	$\sin\alpha$
max	<mi>max</mi>	<ci>max</ci>	maximum	
π	<mi>π</mi>	<ci>π</ci>	Pi	
∈	<mo>∈</mo>	<in/>	belong to	
∫	<mo>∫</mo>	<int/>	integral	

3.2. Edit distance algorithm

In the course of the experiments, it was found that the reason why the content MathML expressions obtained after conversion could not be displayed properly in the web pages was that the positions between symbols and sub-formulas in the expressions crossed. In this study, inspired by the calculation of edit distance and graph edit distance [41,42], the edit distance algorithm is used to generate candidate sets. Edit distance is the minimum number of edit operations required for two sentences to become a unified form, and there are three types of edit operations: add, delete, and replace. For the convenience of calculation, when correcting symbol errors, the contents of the sub-formulae adjacent to the operator are ignored and only the start and end tags of the sub-formulae are retained, and each tag being recorded as a character. This study takes $x+y=z$ as an example to elaborate, as shown in Table 3.

In particular, the "wrong expression" in Table 3 refers to the expression that converted from presentation MathML to content MathML cannot be displayed accurately on the electronic scientific documents and webpage. Therefore, we consider that the content MathML cannot

represent correct semantic information of the corresponding mathematical expression because of the wrong order of the nodes. Content MathML has a strict indentation. Content MathML requires not only that the parameter types of the operators and operands in mathematical expressions are accurate, but also that the order of nodes is correct. So, the wrong expression caused by the wrong order of the nodes requires error correction.

The first column in Table 3 is the presentation MathML that would be converted to content MathML. The second column is the content MathML obtained by using the rule mapping method to convert presentation MathML. But this content MathML cannot be displayed accurately on the electronic scientific documents and webpage because of the wrong order of the nodes. Because this content MathML cannot represent correct semantic information of the corresponding mathematical expression, it requires error correction. The third column is the wrong part of the content MathML that requires to be corrected. The fourth column is the candidate set generated by using the edit distance algorithm for the wrong part of the content MathML. The fifth column is the content MathML obtained by using the Bayesian algorithm error correction. This content MathML can be displayed accurately on the electronic scientific documents and webpage and represent correct semantic information of the corresponding mathematical expression.

Table 3. Error correction process of $x+y=z$.

	1	2	3	4	5
expression	<math> <mrow> <mrow> <mi>x</mi> <mo>+</mo> <mi>y</mi> </mrow> <mo>=</mo> <mi>z</mi> </mrow> </math>	<math> <apply> <apply> <plus/> <ci>x</ci> <ci>y</ci> </apply> <eq/> <ci>z</ci> </apply> </math>	<apply></apply> <eq/> <ci></ci>	<eq/> <apply></apply> <ci></ci> <apply></apply> <ci></ci> <eq/> <eq/> <apply></apply> <cn></cn> <apply></apply> <cn></cn> <eq/> <eq/> <apply></apply> <apply></apply>	<math> <apply> <eq/> <apply> <plus/> <ci>x</ci> <ci>y</ci> </apply> <ci>z</ci> </apply> </math>
interpretation	presentation MathML	content MathML (wrong expression)	the wrong part	candidate sets	content MathML (after error correction)

3.3. Bayesian algorithm

3.3.1. The basic idea of Bayesian formula

The Bayesian error correction algorithm is based on the Bayesian formula. By observing the prior probability of the data and combining it with the conditional probability, the Bayesian formula is used to calculate the posterior probability and select the best data in the candidate set as its error correction object.

The Bayesian formula is defined as follows: This study assumes that in a randomized experiment Q, $h_1, h_2, \cdots, h_n$ are a division of the sample space Ω, where $P(h_i) > 0$ and $i = 1,2\cdots,n$. In addition, D is the observation data in Q and $P(D) > 0$. D only has a corresponding relationship with certain data in the sample space. The Bayesian formula is

$$P(h_i|D) = \frac{P(D|h_i)P(h_i)}{\sum_{i=1}^{n} P(D|h_i)P(h_i)} = \frac{P(D|h_i)P(h_i)}{P(D)} \tag{1}$$

3.3.2. Error correction algorithm

With the Bayesian algorithm in this study, we assume that $R = \{R_1, R_2, \cdots, R_n\}$ is the set of candidate expressions, and $E = \{E_1, E_2, \cdots, E_n\}$ is an incorrect expression with n sub-formulas. The task of the Bayesian error correction algorithm is to compute the most accurate error correction object R_i based on the set of candidate expressions E_i to be corrected. According to Eq (1), the Bayesian error correction algorithm is briefly described as: For any wrong expression E, combined with the Bayesian formula on the basis of prior probability, is calculated to obtain the only accurate expression R_i with the maximum possible error expression E.

According to Eq (1), the Bayesian error correction algorithm is expressed as follows.

$$P(R_i|E) = \frac{P(E|R_i)P(R_i)}{P(E)} \tag{2}$$

The necessary and sufficient conditions that the corrected object of expression E to be corrected is the expression R_i is

$$\frac{P(R_i|E)}{P(R_j|E)} > 1, 1 \le j \le m, i \ne j \tag{3}$$

where $P(R_i \mid E)$ denotes the probability that the expression R_i is the expression E's error correction target. The process of solving for the correct expression can be translated into solving for R_i that maximizes the value of $P(R_i \mid E)$. Since $R_i \in R$ and random E, the following equation is obtained:

$$P(E) = \frac{1}{\Sigma(R_i)} \tag{4}$$

where $\Sigma(R_i)$ is the number of expressions R_i in the data set. That is, no matter which R is the optimal error correction target of E, $P(E)$ is the same value. So, the expression of the Bayesian error correction algorithm can be simplified as

$$P(R_i|E) = P(E|R_i)P(R_i) \tag{5}$$

where $P(R_i)$ represents the prior probability as

$$P(R_i) = \frac{|S_{R_i}|}{|S|} \tag{6}$$

where $|S|$ is the total number of samples of the expressions in the data set and $|S_{R_i}|$ is the number of samples in the data set that contains R_i.

For the calculation of $P(E|R_i)$, if there are multiple parts of the expressions to be corrected that require error correction, in order to reduce the time and space complexity of the calculation process. In particular, assuming that the conditions are independent of each other, it can be described as shown in Eq (7).

$$P(E|R_i) = \prod_{k=1}^{n} P(E_k|R_i) = P(E_1, E_2, \cdots, E_n|R_i) = P(E_1|R_i)P(E_2|R_i)\cdots P(E_n|R_i) \tag{7}$$

The value of $P(E_k|R_i)$ can be derived by training the sample expressions in the data set, according to the statistics shown in Figure 2. This study assumes that the expressions are continuous attributes obeying Gaussian distribution, they can be described as shown in Eq (8). The $f(\chi;\mu,\sigma)$ is the Gaussian density function, μ is the average value, and σ is the standard deviation.

$$P(E_k|R_i) = f(\chi;\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(\chi-\mu)^2}{2\sigma^2}\right) \tag{8}$$

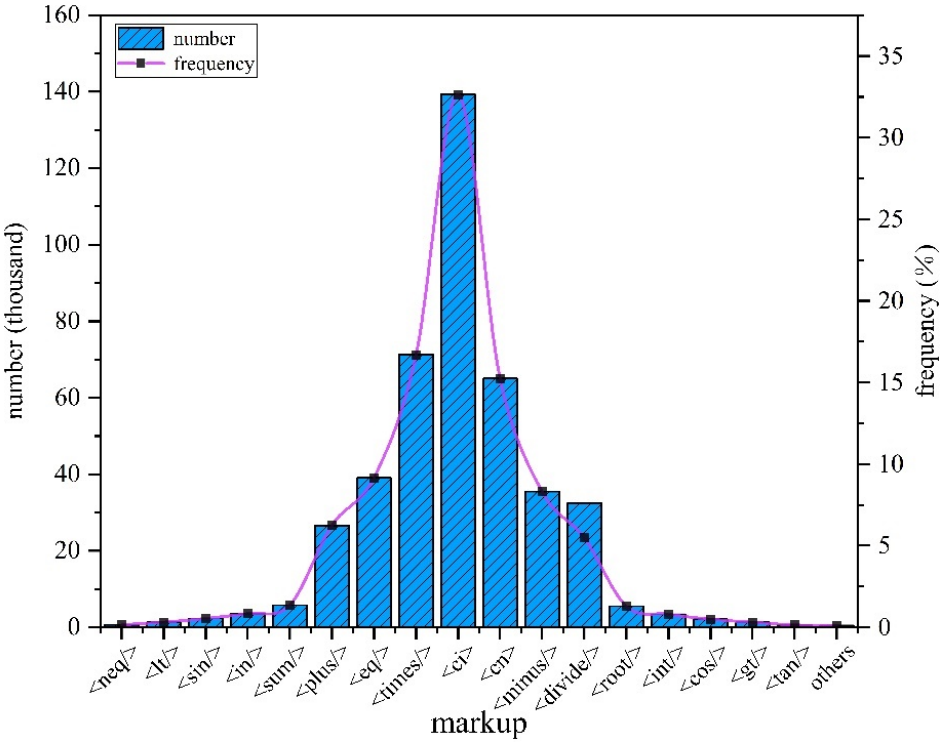

Figure 2. The statistic of sample expressions.

In summary, for the sample expression E the value of $P(E|R_i)P(R_i)$ is calculated and the expression with the highest probability is the corrected object of E. It can be described as shown in Eq (9). To reduce the calculation error, this study takes the logarithm for Eq (9), as shown in Eq (10).

$$R = \arg\max P(R_i)\prod_{k=1}^{n} P(E_k|R_i) \tag{9}$$

$$R = \arg\max\left[\ln P(R_i) + \sum_{k=1}^{n} \ln P(E_k|R_i)\right] \tag{10}$$

An expression usually contains at least one mathematical symbol. Therefore, in the process of correcting mathematical expressions, it is necessary to check the symbols in the expressions one by one and then correct them. First, the expression is selected from the data set as the expression to be corrected. Then, the expression is traversed to locate the symbol tag and find the two sub-expressions adjacent to is so that it is a part to be corrected. Next, the candidate set is generated using the edit distance algorithm with the symbols to be corrected as the target and the expressions in the data set. Finally, the result is calculated and sorted according to the Bayesian formula and the best one is selected. These steps are repeated until all symbols in the error expression are corrected. The general process is shown in Algorithm 2.

Algorithm 2. The Bayesian error correction algorithm of expression semantic.

INPUT: F_e
OUTPUT: F_r

```
F, I
iterate(F_e) ← S_i
 while F_e do
   if S_i != null && i<n
     S_i && DataSet
     Set{Formula}  // candidate set
     F_i = max[Bayes(Set)] calculate the probability
     i += 1  //
   else
     F_r ← F_e
 end while
return F_r
```

4. Experimental results and analysis

4.1. Experimental data and environment

The experiment is implemented in the JDK (Java Development Kit) 1.8 environment. The Eclipse platform and the Java language are used to program the experiment. The experimental environment is shown in Table 4.

Table 4. Experimental environment.

Experimental environment	Configuration
Processor	Intel(R) Core (TM) i5-8500, 3.00GHz
Operating system	Microsoft Windows 10
Development tool	Eclipse, Java
Video memory	8G

The public data set Ntcri-Mathir-Wikipedia-Corpus is used as the training set in this study. This data set contains 11,792 documents and 124,878 expressions have been extracted, which are used to establish mapping rules and build a data dictionary for the experiment. The test set used in the experiment is 78,348 expressions extracted from 10,372 Chinese documents collected by the laboratory from the CNKI website.

4.2. Experiment result and data analysis

In order to verify the effectiveness of the algorithm in this paper quickly and efficiently, an unduplicated random sample of any 5000 mathematical expressions are selected and divided into 10 groups for the experiment. The experimental results are analyzed from three aspects: the conversion result of the rule mapping method, the performance of the Bayesian error correction algorithm, and the time efficiency of the algorithm.

4.2.1. Evaluation metrics

In order to present the experimental results clearly, the comprehensive evaluation metrics F-Measure is used in this study to evaluate the experimental results. F-Measure is the weighted harmonic average of precision P and recall R, as shown in Eq (11):

$$F=\frac{\left(\alpha^2+1\right)*P*R}{\alpha^2*\left(P+R\right)} \tag{11}$$

The most common F_1 with $\alpha=1$ is adopted in this study, as shown in Eq (12). The higher the F_1, the better the performance. The precision P and recall R are calculated as shown in Eqs (13) and (14) respectively, where N denotes the total number of expressions in the current sample group. TN denotes the total number of expressions that can be accurately represented in the web page by the semantic expressions obtained by the rule mapping method, i.e., accurate expressions. FN denotes the total number of expressions that are not accurately represented in the web page by the semantic expressions obtained by the rule mapping method, i.e., the expressions to be corrected. T_{FN} denotes the total number of expressions that can be accurately represented in the web page after Bayesian error correction. The error correction rate is represented by cr. The specific equations are shown as follows.

$$F_1=\frac{2*P*R}{P+R} \tag{12}$$

$$P=\frac{TN}{TN+FN} \tag{13}$$

$$R=\frac{TN}{N} \tag{14}$$

$$cr=\frac{T_{FN}}{FN} \tag{15}$$

4.2.2. Conversion result and analysis of the rule mapping method

This study conducts experiments on 10 sets of sample expressions respectively. The rule mapping method is used to convert presentational expressions into semantic expressions and the number of expressions that can be converted is recorded. Semantic expressions are tested manually whether they can be presented as the corresponding mathematical expressions in the web page. If they can be presented as the corresponding mathematical expressions, they are accurate expressions. Otherwise, they are expressions that need to be corrected.

From Table 5, it can be seen that not all presentational expressions can be converted to semantic expressions using the rule mapping method. The reason is that in presentational expressions, irregular expressions can be represented as presentation MathML, such as poorly formatted expressions $xy+$, $x-$, and so on. However, these incomplete expressions cannot be encoded as content MathML in semantic expressions. Therefore, with the rule mapping method, they cannot be fully converted.

Table 5. The results of conversion.

Group	1	2	3	4	5	6	7	8	9	10
Sample expressions	500	500	500	500	500	500	500	500	500	500
Converted expressions	495	489	491	490	485	494	489	490	490	493
Correct expressions	124	117	121	113	109	128	106	115	122	127
Wrong expressions	371	372	370	377	376	366	383	375	368	366

4.2.3. Performance analysis of the Bayesian error correction algorithm

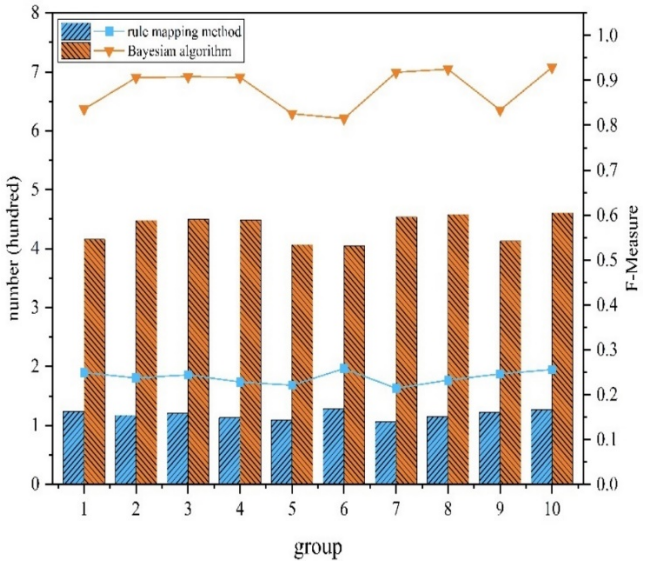

Figure 3. Comparison of error correction results.

The comparison results of the number of accurate expressions and F-measures of the rule mapping method and the Bayesian algorithm are shown in Figure 3. The error correction rate of the Bayesian algorithm on 10 sets of experimental data is shown in Table 6.

Table 6. Error correction rate.

Group	1	2	3	4	5	6	7	8	9	10	Average
Sample expressions	500	500	500	500	500	500	500	500	500	500	-
Wrong expressions	371	372	370	377	376	366	383	375	368	366	-
Corrected expressions	292	331	329	336	298	277	348	343	291	334	-
Error correction rate	0.787	0.890	0.889	0.891	0.792	0.757	0.909	0.915	0.791	0.913	0.853

It can be seen from Figure 3 that a certain number of accurate expressions can be directly obtained by the rule mapping method, because there are some simple expressions in the data set, such as $+$, $-$,

*, and ÷, from which the rule mapping method can directly get accurate expressions.

The experimental data of the first, fifth, sixth, and ninth groups show that the error correction rate is lower than the average, and the F_1 value after the Bayesian error correction algorithm is also lower than those of the other groups. The main reasons are as follows. First, the data set contains some expressions with special symbols, such as the *bra*−*ket* symbolic. The symbol is denoted by $\langle\phi|\psi\rangle$, consisting of a left part $\langle\phi|$ called the bra, and a right part $\langle\psi|$ called the ket. In the process of conversion and error correction, it is not possible to obtain the corresponding accurate semantic expressions based only on the content information in the presentational expressions. Second, a symbol can represent multiple meanings, such as ( ) which can represent an interval $(-\pi,\pi)$, a vector (x,y), a binomial $\binom{n}{k}$, and so on. Their content markups are <interval>, <vector>, and <binomial>. Therefore, a certain error may occur during error correction, and the process of error correction may fail. Third, due to the insufficient number of expressions in the data set, the statistical results are biased and the final results are affected.

4.2.4. Analysis of response time

In this study, the presentational expressions in Chinese electronic scientific documents are converted to semantic expressions with semantic information by using rule mapping method. However, using only the rule mapping method can cause problems such as incorrect structure of mathematical expressions. Therefore, the Bayesian error correction method is designed to correct the obtained wrong expressions. The response time of the two methods is shown in Figure 4. It can be seen from Figure 4 that the response time of the system increases as the number of expressions increases. Since the Bayesian error correction algorithm is a step after the rule mapping method, it causes an increase in response time. However, the increase in time complexity is not large and is still within an acceptable range. Therefore, this method is considered reasonable given the premise of improving the accuracy of the experimental results.

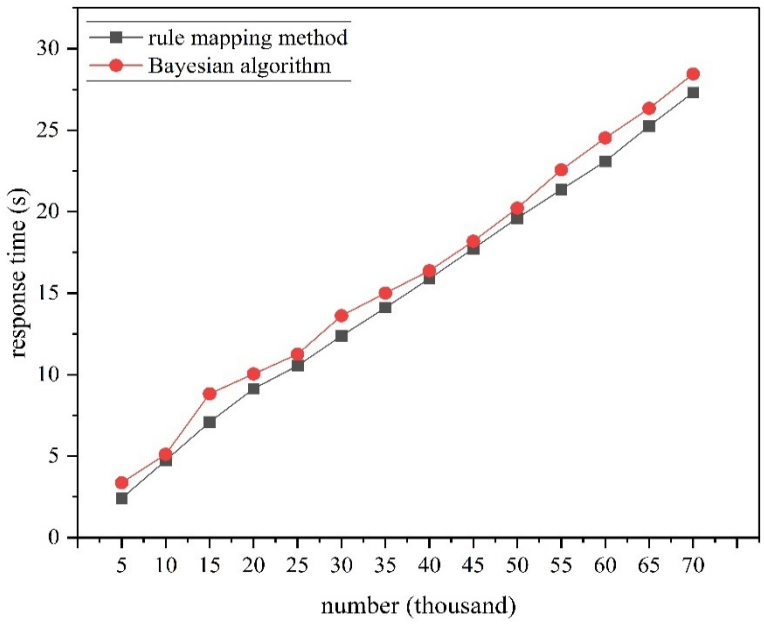

Figure 4. Response time of the algorithm.

5. Conclusions and future works

This study addresses the problem that the expressions with semantic information obtained by using only the rule mapping method are prone to errors, this study proposes a Bayesian algorithm to perform semantic errors correction on the converted expressions. In the semantic error correction process, the candidate sets are obtained by combining the edit distance algorithm. And then, the probability of each expression in the candidate set is calculated by Bayesian formula. Finally, these expressions are sorted to get the best result.

The presentational expressions and semantic expressions in the data set NTCIR are used as the training set in this study. Based on the markups and syntactic structure of the two formats, the mapping rules for the expression format conversion are formulated, and the parameters of the Bayesian error correction algorithm are statistically trained. The presentational expressions in the data set collected by the laboratory are used as the test set to test the performance of the algorithm. The experimental results prove that the average F_1 value of the method using only the rule mapping is 0.239, and the average F_1 value of the method using Bayesian error correction is 0.881, which is a significant improvement over the former. The average error correction rate cr is 0.853, indicating that the Bayesian error correction algorithm can effectively correct the semantic of expressions. Since the Bayesian algorithm is performed on the basis of the rule mapping method, the response time increases. But the increase is within an acceptable range. Therefore, under the premise of improving the accuracy of the experimental results, this method is reasonable and the experiment has research significance.

Since presentational expressions focus more on the content in the expression and display only the symbols. Semantic expressions focus more on the inner meaning of the expression and need to display mathematical information. From the experimental, it is clear that the ambiguity of the symbols may have some influence on the results. For example, “e” in expressions have different semantics in different situations. It can represent either a variable or a mathematical constant in an expression. To solve the problem of semantic ambiguity of mathematical expressions, in the next step of our study, on the one hand, we will try to disambiguate expressions by combining the relevant textual context of the mathematical expression; on the other hand, we will try to expand the content dictionaries to correctly define more complex functions. In addition, when the Bayesian formula is used for calculation, the statistical result of the expression content in the data set is required. Therefore, to make the experimental results more robust, the experimental data set needs to be enriched.

Acknowledgments

This work was supported by Science and Technology Project of Hebei Education Department (ZD2019131), the Natural Science Foundation of Hebei Province (F2019201451) and "One province, one university" fund of Hebei University (521000981155).

Conflict of interest

The authors declare no conflict of interest.

References

1. P. Amarnath, P. Partha, G. Alexander, A formula embedding approach to math information retrieval, *Comput. Y Sistemas*, **22** (2018), 819–833. https://doi.org/10.13053/CyS-22-3-3015
2. T. Chih-Fong, K. Shih-Wen, M. Kenneth, M. Y. Lin, LocalContent: A personal scientific document retrieval system, *Electr. Lib.*, **33** (2015), 373–385. https://doi.org/10.1108/EL-08-2013-0148
3. W. Zhong, S. Rohatgi, J. Wu, C. L. Giles, R. Zanibbi, Accelerating substructure similarity search for formula retrieval, in *Proceedings of the European Conference on Information Retrieval*, (2020), 714–727. https://doi.org/10.1007/978-3-030-45439-5_47
4. B. Mansouri, S. Rohatgi, D. W. Oard, J. Wu, R. Zanibbi, Tangent-CFT: an embedding model for mathematical formulas, in *Proceedings of the ACM SIGIR International Conference on the Theory of Information Retrieval (ICTIR)*, 2019. https://doi.org/10.1145/3341981.3344235
5. S. Dhar, A. Biswas, N. Singh, SciMath: A mathematical information retrieval system using signature based B tree indexing, *Int. J. Innovat. Technol. Explor. Eng.*, **8** (2019), 234–244. https://doi.org/10.35940/ijitee.K1298.0981119
6. Y. Nagao, N. Suzuki, Classifying mathML expressions by multilayer perceptron, *IEICE Trans. Inf. Syst.*, **E101** (2018), 1954–1958. https://doi.org/10.1587/transinf.2017edl8211
7. Y. P. Qin, J. N. Guo, A. H. Zhang, A novel extreme learning fault diagnosis based supervision applied to mathematical formula contrastive analysis, *Neurocomputing*, **177** (2016), 166–273. https://doi.org/10.1016/j.neucom.2015.11.027
8. P. Sojka, M. Líška, M. Růžička, Building corpora of technical texts : Approaches and Tools, in *the Proceedings of the Fifth Workshop on Recent Advances in Slavonic Natural Languages*, 2011. Available from: https://www.fi.muni.cz/usr/sojka/papers/sojka-liska-ruzicka-raslan2011.pdf.
9. M. Růžička, P. Sojka, M. Líška, Math indexer and searcher under the hood: history and development of a winning strategy, in *Proceedings of the 11th NTCIR Conference*, 2014. Available from: http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings11/pdf/NTCIR/Math-2/07-NTCIR11-MATH-RuzickaM.pdf.
10. N. Kando, T. Sakai, C. Clarke, NTCIR (NII Testbeds and Community for Information access Research) Project, 2016. Available from: http://research.nii.ac.jp/ntcir/index-en.html.
11. Tsinghua University, Ltd., CNKI (China National Knowledge Infrastructure). https://www.cnki.net.
12. T. Zhang, L. Li, W. Su, Y. J. Zhao, A mathematical formulae converter based on Math Edit, *Comput. Appl. Software*, **27** (2010), 14–16. https://doi.org/10.3969/j.issn.1000-386X.2010.01.006
13. H. Sharaf, B. Samita, K. Shakeel, Rule based conversion of LaTeX math equation into Content MathML (CMML), *J. Inf. Sc. Eng.*, **36** (2020), 1021–1034. https://doi.org/10.1109/ICSCC.2019.8843592
14. S. Y. Zhu, L. Hu, R. Zanibbi, Rotation-robust math symbol recognition and retrieval using outer contours and image subsampling, in *Proceedings of Society of Photo-optical Instrumentation Engineers (SPIE)*, 2013. https://doi.org/10.1117/12.2008383
15. W. Su, Research on web-based input and accessibility of mathematical expressions, 2010. Available from: http://cdmd.cnki.com.cn/article/cdmd-10730-1011034166.htm.
16. M. Schubotz, A. Grenier-Petter, P. Scharpf, N. Meuschke, H. Cohl, B. Gipp, Improving the representation and conversion of mathematical formulae by considering their textual context, in *Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries (JCDL)*, 2018. https://doi.org/10.1145/3197026.3197058

17. C. Cai, W. Su, L. Li, On key issues of converting presentation mathematics formulas to content, *Comput. Appl. Software*, **29** (2012), 30–33. https://doi.org/10.3969/j.issn.1000-386X.2012.08.008
18. I. A. Doush, F. Alkhateeb, E. A. Maghayreh, Towards meaningful mathematical expressions in e-learning, in *Proceedings of the 1st International Conference on Intelligent Semantic Web-Services and Applications*, 2013. https://dl.acm.org/doi/pdf/10.1145/1874590.1874612
19. M. Nghiem, G. Y. Kristianto, A. Aizawa, Using mathML parallel markup corpora for semantic enrichment of mathematical expressions, *Ieice Trans. Inf. Syst.*, **96** (2013), 1707–1715. https://doi.org/10.1587/transinf.E96.D.1707
20. I. Toloaca, M. Kohlhase, Notation-based semantification, in *Conference on Intelligent Computer Mathematics*, 2016. Available from: http://ceur-ws.org/Vol-1785/M6.pdf.
21. A. Greiner-Petter, M. Schubotz, H. Cohl, B. Gipp, Semantic preserving bijective mappings for expressions involving special functions in computer algebra systems and document preparation systems, *Aslib J. Inf. Manage.*, **71** (2019). https://doi.org/10.1108/AJIM-08-2018-0185
22. M. Grigore, M. Wolska, M. Kohlhase, Towards context-based disambiguation of mathematical expressions, *Asian Symp. Comput. Math. Math. Aspects Comput. Inf. Sci.*, 2009. Available from: https://kwarc.info/people/mkohlhase/papers/ASCM-DML09.pdf.
23. A. K. Nketia, W. H. Tian. Toward perfect neural cascading architecture for grammatical error correction, *Appl. Intell.*, **51** (2021), 3775–3788. https://doi.org/10.1007/s10489-020-01980-1
24. S. Li, J. B. Zhao, G. R. Shi, Y. P. Tan, H. F. Xu, G. Chen, Chinese grammatical error correction based on convolutional sequence to sequence model, *IEEE Access*, **7**(2019), 72905–72913. https://doi.org/10.1109/ACCESS.2019.2917631
25. H. Daniel, S. Jan, P. Matus, Survey of automatic spelling correction, *Electronics*, **9** (2020). https://doi.org/10.3390/electronics9101670
26. Y. E. Jing, Analysis of grammar error correction algorithm based on deep learning technology, *Inf. Technol.*, **9** (2020), 143–148. https://doi.org/CNKI:SUN:HDZJ.0.2020-09-031
27. J. M. Ye, D. X. Luo, S. Chen, A text error correction model based on hierarchical editing framework, *Acta Electr. Sinica*, **49** (2021), 401–407. https://doi.org/10.12263/DZXB.20200448
28. J. X. Gu, B. Yang, Survey on Bayesian optimization methodology and application, *J. Software*, **29** (2018), 3068–3090. https://doi.org/10.13328/j.cnki.jos.005607
29. M. U. Sadiq, M. M. Yousaf, L. Aslam, M. Aleem, S. Sarwar, S. W. Jaffry, NvPD: novel parallel edit distance algorithm, correctness, and performance evaluation, *Cluster Comput. J. Netw. Software Tools Appl.*, **23** (2020), 879–894. https://doi.org/10.1007/s10586-019-02962-w
30. G. Z. Sun, J. W. Lv, H. K. Li, MeTCa: Multi-entity trusted confirmation algorithm based on edit distance, *Comput. Sci.*, **47** (2020). https://doi.org/10.11896/jsjkx.191100176
31. P. Ni, J. Li, H. Hao, Q. Han, X. Du, Probabilistic model updating via variational Bayesian inference and adaptive Gaussian process modeling, *Comput. Methods Appl. Mechan. Eng.*, **383** (2021). https://doi.org/10.1016/j.cma.2021.113915
32. J. Zhao, X. Liu, S. Sun, Probabilistic inference of Bayesian neural networks with generalized expectation propagation, *Neurocomputing*, **412** (2020), 392–398, https://doi.org/10.1016/j.neucom.2020.06.060
33. A. Rahman, U. Qamar, A Bayesian classifiers based combination model for automatic text classification, in *Proceedings of the 7st IEEE International Conference on Software Engineering and Service Science*, (2016), 63–67. https://doi.org/10.1109/ICSESS.2016.7883016

34. Y. Qussai, J. Yaser, K. N. Viet, An evaluation and analysis of static and adaptive Bayesian spam filters, *J. Int. Technol.*, **19** (2018), 1015–1022. https://doi.org/10.3966/160792642018081904005
35. J. Liu, Z. Wang, H. Wang, Research on spam filtering technology based on IMI-WNB algorithm, *Comput. Eng.*, **46** (2020), 299–305. https://doi.org/10.19678/j.issn.1000-3428.0056577
36. A. N. Ngaffo, E. A. Walid, C. Zied, A Bayesian inference based hybrid recommender system, *IEEE Access*, **8** (2020). 101682–101701. https://doi.org/10.1109/ACCESS.2020.2998824
37. F. Y. Liu, X. Q. Gao, Z. Zhang, Improved Bayesian probabilistic model based recommender system, *Comput. Sci.*, **44** (2017). https://doi.org/10.11896/j.issn.1002-137X.2017.05.052.
38. M. L. Zhan, L. Roger, K. Andrew, Pronoun interpretation in Mandarin Chinese follows principles of Bayesian inference, *Plos One*, **15** (2020). https://doi.org/10.1371/journal.pone.0237012
39. X. Yi, Y. U. Chen, Y. Shi, Bayesian method for intention prediction in pervasive computing environments, *Scientia Sinica (Informationis)*, 2018. Available from: Available from: http://en.cnki.com.cn/Article_en/CJFDTotal-PZKX201804006.html.
40. K. Jebran, L. S. Chang, Enhancement of sentiment analysis by utilizing noisy social media texts, *J. Korean Inst. Commun. Inf. Sci.*, **45** (2020), 1027–1037. https://doi.org/10.7840/kics.2020.45.6.1027
41. K. Chatterjee, T. A. Henzinger, R. Ibsen-Jensen, J. Otop, Edit distance for pushdown automata, in *Inrernational Coloquium* on *Automata, Languages, and Programming*, (2015), 121–133. https://doi.org/10.1007/978-3-662-47666-6_10
42. R. Romain, On the unification of the graph edit distance and graph matching problems, *Pattern Recognit. Lett.*, **145**(2021), 240–246. https://doi.org/10.48550/arXiv.2104.06186

AIMS Press