Smart Learning Environments

RESEARCH Open Access

Automated labeling of PDF mathematical exercises with word N-grams VSM classification

Taisei Yamauchi[1], Brendan Flanagan[2*], Ryosuke Nakamoto[1], Yiling Dai[3], Kyosuke Takami[4] and Hiroaki Ogata[3]

*Correspondence:
flanagan.
brendanjohn.4n@kyoto-u.ac.jp

[1] Graduate School of Informatics, Kyoto University, Kyoto, Japan
[2] Center for Innovative Research and Education in Data Science, Institute for Liberal Arts and Sciences, Kyoto University, Kyoto, Japan
[3] Academic Center for Computing and Media Studies, Kyoto University, Kyoto, Japan
[4] Education Data Science Center, National Institute for Educational Policy Research, Tokyo, Japan

Abstract

In recent years, smart learning environments have become central to modern education and support students and instructors through tools based on prediction and recommendation models. These methods often use learning material metadata, such as the knowledge contained in an exercise which is usually labeled by domain experts and is costly and difficult to scale. It recognizes that automated labeling eases the workload on experts, as seen in previous studies using automatic classification algorithms for research papers and Japanese mathematical exercises. However, these studies didn't delve into fine-grained labeling. In addition to that, as the use of materials in the system becomes more widespread, paper materials are transformed into PDF formats, which can lead to incomplete extraction. However, there is less emphasis on labeling incomplete mathematical sentences to tackle this problem in the previous research. This study aims to achieve precise automated classification even from incomplete text inputs. To tackle these challenges, we propose a mathematical exercise labeling algorithm that can handle detailed labels, even for incomplete sentences, using word n-grams, compared to the state-of-the-art word embedding method. The results of the experiment show that mono-gram features with Random Forest models achieved the best performance with a macro F-measure of 92.50%, 61.28% for 24-class labeling and 297-class labeling tasks, respectively. The contribution of this research is showing that the proposed method based on traditional simple n-grams has the ability to find context-independent similarities in incomplete sentences and outperforms state-of-the-art word embedding methods in specific tasks like classifying short and incomplete texts.

Keywords: Automatic labeling, Word n-gram, Random forest, Incomplete text classification, Word embedding, Mathematical education, Mathematical education in Japan

Introduction

Labeling learning materials is a key problem in scaling smart learning environments (Contractor et al., 2015). The availability of knowledge metadata for learning materials is critical as important decisions, such as what to recommend for study for the next time, are usually made based on the metadata and the learners' previous experience (Vovides et al., 2007). Each exercise in a textbook for each subject usually has a set of course units that clarify the category of each exercise and are very useful in educational situations and

the framework of educational problems. Recently, there has also been a growing trend in the adoption of nationwide curriculum or studying guidelines, such as: the Australian digital curriculum called Australian Curriculum, Assessment and Reporting Authority (ACARA) in Australia (Ditchburn, 2012), Common Core Standards (Porter et al., 2011; Ritter, 2009) in America, Mathematics Curriculum Standards for Compulsory Education (MOE, 2012) in China, and the Courses of Study (MEXT, 2018) in Japan. These guidelines provide regulations for education and instruction, as well as standard units for each subject (MEXT, 2018). Educators select learning materials based on these guidelines to meet the requirements of the compulsory curriculum. Therefore, learning materials that do not contain knowledge metadata are difficult to incorporate into the course of study, and the automated assignment of labels to learning materials could help overcome this problem.

In this study, the task of labeling learning materials has two main objectives: yielding high accuracy for detailed classification and labeling incomplete texts. First, as is common with other labeling tasks, the performance of the classification task is very important as the aim of labeling materials is to reduce the burden of domain experts who are usually manually tackling the knowledge classification task. Detailed labeling of learning materials is very useful in the educational field, but assigning the classification to problems manually is a hard task that requires the cooperation of experts, and the burden could be alleviated through automation. Schubotz et al. (2020), examined the task of automatically assigning coarse labels according to a mathematical subject classification scheme for retrieving research papers and literature on mathematics in English. It was found that the support provided by the proposed automatic classification algorithm resulted in a reduced manual classification burden for domain experts. Another study proposed the WE-KE model, which combines word embedding and knowledge components, to achieve accurate unit classification of Japanese mathematical exercises (Tian et al., 2022). With the shift to ICT education, researchers label the exercises to utilize them for learning pattern analysis (Wang et al., 2022). While more detailed classifications may be necessary depending on the intended use, such detailed labeling was not conducted in those studies.

Second, as extracting complete text is sometimes difficult due to the format of learning materials, another approach for labeling incomplete text is required. With the increased digitization of learning materials and their use in smart learning environments, teachers and publishers are migrating existing non-digital materials to these systems. As these learning materials were usually not created while considering digitization, it is often seen that publishers will provide publication-quality PDFs directly to teachers or educational institutes. Problems are caused when uploading and analyzing such materials in learning environments as it is difficult to extract all of the information, such as: text, formulas, graphs, and images, from publication-quality PDFs, resulting in incomplete information extraction (Abekawa & Aizawa, 2016). While researchers have tried labeling with sentences, images, formulas, or a combination of them (Bhartiya et al., 2016; Shen, et al., 2021; Tian et al., 2022; Wang et al., 2022), there has been less focus on classification with incomplete information of mathematical sentences. In this study, we propose a mathematical exercise labeling algorithm that can deal with detailed labels, even for incomplete sentences, by focusing on the exact match of a set of mathematical exercises and

predicting a unit using an existing machine learning method or calculating the similarity of any given exercise to a set of weighted word n-grams. Therefore, we aim to answer the following research question:

RQ: What are the best features and models that can assign detailed and precise labels from incomplete mathematical exercise text?

We propose an algorithm to automatically provide classification results for preprocessed exercise sentences that have been extracted from publication-quality PDFs that include incomplete text. In the experiments of this study, two different levels of labels are assigned to each exercise for validation. We then predict the labels to evaluate the performance of the proposed algorithm and compare it to state-of-the-art word embedding models.

Literature review

Labeling learning materials

National labeling standards for mathematical exercises

Often learning materials are labeled to notice easily what kind of knowledge is contained in an exercise. Government standards often provide some norms of mathematical exercise classification, for example, in Japan the government provides common standards of subjects and directions for each unit of study that aim to develop the qualities and abilities to think mathematically through mathematical activities in the Guidelines for the Course of Study for Senior High Schools (MEXT, 2009, 2018), and teachers prepare exercises by following these directions. In the US, the Common Core State Standards (CCSS) classification refers to the learning standards for K-12 education that was developed in collaboration with teachers, school administrators, and professionals to provide a clear and consistent framework for preparing children for college and career success (Ritter, 2009). It includes 11 units that students will study over the course of nine years, plus appendices that cover counting and radix, operations and algebraic thinking, decimal numbers and operations, fraction operations, measurement and data, ratios and proportion relationships, number systems, expressions and equations, functions, geometry, statistics, and probability, as well as content taught in higher grades (Shintani, 2014). In Mathematics Curriculum Standards for Compulsory Education in China (MOE, 2012), learning items are distributed into one of up to four main parts and assigned categories 10 keywords, including: number sense, symbolic awareness, space concept, geometry intuitive, data analysis concept, computation ability, reasoning ability, model idea, application awareness, and innovative awareness (Guo et al., 2018). In Australian Curriculum, Assessment and Reporting Authority, treated as an Australian digital curriculum (Ditchburn, 2012), units are called "content strands" and consist of number and algebra, measurement and geometry, and statistics and probability. Each of these strands has 6, 5, and 2 units, respectively, and the structure can be described as hierarchical (ACARA). There is also a specialized system called Zentralblatt MATH (zbMATH) that is a mathematics-related bibliographic database and literature search engine. The Mathematics Subject Classification (MSC) which zbMATH helps maintain is used to classify items in mathematical sciences literature. Every 10 years, two editorial groups solicit input from the

mathematical community. The new MSC (MSC2020) includes 63 two-digit classifications, 529 three-digit classifications, and 6006 five-digit classifications (Dunne & Hulek, 2020; Kühnemund, 2016).

As the topic standards mentioned above can be important rules when classifying many mathematical materials, some researchers decided to tackled labeling math exercises based on the standard automatically. One study has attempted to classify according to the CCSS (Ritter, 2009), and this study used 385 different labels to classify 12 years of mathematics materials from kindergarten through to high school (Shen et al., 2021). One study also proposed the MathBERT model (Shen et al., 2021), which is a model created by preparing a large mathematical corpus ranging from the pre-kindergarten to the graduate level and training a base BERT model (Devlin et al., 2019). However, these studies did not tackle the problem of incomplete text classification. In this study, we use information from MEXT to label exercise data in both a coarse and detailed method while focusing on incomplete exercise text labeling.

Labeling for analysis of how students learn

There is a trend toward analyzing learning behavior in a new way using labels assigned to teaching materials. Regarding the use of features in the analysis of learning effectiveness, a study reported that the proposed system automatically assigned labels with learning materials and the study shows the assigned labels can assist in the discovery of students' learning patterns (Wang et al., 2022). While the analysis using labels is novel in the research, the labeling conducted in this research was only for one class in the university and was not generalized using a common standard.

Giving labels to exercises for knowledge tracing is also a hot research topic. One study, using multiple real data sets consisting of tens of thousands of users and items, showed that regression classification models could accurately and rapidly estimate student knowledge, even when student data is sparsely observed. In addition, the study showed that the model can handle multiple knowledge elements and side information such as the number of trials of items and skill levels (Vie et al., 2019). If no labels were given to each exercise, the study could not accurately predict the student's performance.

It is also useful to categorize any exercise for recommending a specific exercise to enhance students' understanding. One study discusses the application of a topic-based tree structure to personalized adaptive educational systems for its transparency for the users (Sosnovsky & Brusilovsky, 2015). Another study focuses on the visualization of the relationship between any combination of two topics to notify the achievements of each student individually, which aims to be consistent among the assessments in different courses, to do meaningful feedback to individual, and to grasp the students' long-term progress (Khosravi & Cooper, 2018). There has also been research into extracting labels from learning materials to form knowledge structure representations that learners can use to increase their awareness of the study process (Flanagan et al., 2019). These research examples show that it is easier to obtain or utilize detailed information about the characteristics of the material if they are labeled in advance. In addition, there is one system, called BookRoll, that any learner can post the PDF materials freely without selecting any topics (Flanagan & Ogata, 2018), so in this context the automatic labeling system helps the materials to obtain some topics.

In this study, we tackle the task of text classification to automate knowledge labeling process for incomplete text by proposing a more detailed and highly accurate method based on n-grams. The proposed method could improve the use of materials with knowledge labeling and assist in the analysis of how students study using these materials.

Labeling to reduce the burden on domain experts

Automatic labeling and classification of learning materials is a prominent area of classification research in education. Schubotz et al., (2020), proposed an automatic classification method in a mathematical subject classification scheme for organizing mathematical literature, achieving a classification agreement rate of 81% with very close accuracy in two large peer-review services. It also enabled an 86% reduction in labor when compared to the manual classification task. The result shows the advantage of labeling automatically, although the research has a different context when compared to the present paper. Tian et al. (2022), proposed a unit classification method that combines natural language processing techniques with a method for extracting keywords from mathematical exercises, and this resulted in a 25% labor reduction compared to manual classification. While the paper provides a mostly accurate classification of units, it only provides as detailed a classification as the Courses of Study even though more detailed labeling may be necessary depending on the intended use.

Automated detailed labeling must be accurate in order to reduce the burden on domain experts and assist in assigning labels to exercises. In this study, we developed a more detailed automated classification that has high accuracy even when labeling exercises that contain incomplete text.

Hierarchical and automatic labeling of teaching materials

Hierarchical text classification (HTC) is a method that can classify objects into multilevel detailed classifications, and this aims to assign one or more optimal categories to text documents from a hierarchical category space (Graovac, 2017) and literature in this area has applied this method to many different types of domains (Silla & Freitas, 2011). Another study proposes a method of categorizing and labeling educational materials with various academic learning objectives (Bhartiya et al., 2016). This method selected words in the materials as labels and achieved extensive labeling in various grades and subjects.

When labeling the exercises, the granularity that is required depends on how the labels will be used, so by assigning different labels to each exercise the scope of use can be broadened. In the experiments, we assigned two labels to each exercise, such as: 1st level unit and 2nd level unit and measured the classification accuracy of each label. Previous studies related to labeling materials for use in Japanese schools don't consider the hierarchical label. Tian et al. (2022) uses 24 labels for the Japanese high school curriculum, and Wang et al. (2022) uses 47 for a course at a university in Japan. Our study uses the most detailed labeling scheme of all previous studies into Japanese mathematical exercise classification with a total of 297 items at the 2nd unit level.

Text vectorization method for classification tasks

N-gram

We often use text mining, machine learning, and natural language processing to classify many kinds of text data, such as: electronic documents, online news, blogs, emails, and digital libraries, to obtain meaningful knowledge, and many classification methods have been proposed (Khan et al., 2010). Previously, Suen (1979) showed that n-gram classification is effective to classify incomplete sentences from OCR. Text classification must work reliably for all input, and therefore must allow some tolerance for various types of text error problems, such as misspellings and grammatical errors in e-mail and character recognition errors in OCR-processed documents, and Cavnar and Trenkle (1994) argued that n-grams is an effective way to meet this requirement. Graovac (2014) proposed an n-gram method for topic-based text classification using the characters in a text so that the method is independent of language and topic.

The task of classification using n-grams has been investigated in various studies. A study on the results of using an n-gram-based algorithm for Bangla text classification (Mansur, 2006) and a study that attempted to statistically estimate the expressive quality of an article by using word n-grams and part-of-speech n-grams in the article (Kobayashi et al., 2012). Despite the loss of semantic information, bag-of-n-grams-based methods have been shown to perform well in sentiment analysis (Li et al., 2016). Many studies have also found n-grams to be an effective tool for classification tasks in a variety of fields, such as in music analysis (Zheng et al., 2017).

However, there are still few studies that use n-gram to classify Japanese mathematical exercise materials. Our study uses n-grams and applies it as a novel method of Japanese mathematical text classification.

Word embedding

Recently, word embedding methods have become a popular text vectorization method, and one of the most representative and popular word embedding methods is Word2Vec (Mikolov et al., 2013). This method trains a model on context-independent distributed representations for words. Considering the context of the sentence using RNN or LSTM, machine learning improves the understanding of sentences, such as: ELMo (Peters et al., 2018) that uses LSTM for a contextualized word embedding model. Moreover, OpenAI's GPT model (Radford et al., 2019) is a model that can have enhanced flexibility for fine-tuned tasks, which allows an AI to consider words at a distance and to compute it not as a Markov method, but in parallel. BERT (Devlin et al., 2019) is also a popular natural language model created by Google which has an attention mechanism instead of RNN and applies a masked language model for learning.

Prior studies have demonstrated the efficacy of word embedding for label classification tasks. For instance, Dharma et al. (2022) utilized the Fasttext method to classify a dataset of 19,977 news articles and 20 news topics with 97.2% accuracy, outperforming other word embedding techniques. However, in the case of short sentence exercises, the sentence vectorization methods using word embedding has been found to be less effective. Tian et al. (2022) applied word embedding for the classification of short Japanese exercise texts, achieving an accuracy of 72.87%. The combination of this method along with the extraction of keywords, called the WE-KE model, further enhanced the

accuracy to 79.57%. These findings suggest that word embedding may not be as effective for short exercise texts. It is worth noting that for this experiment, incomplete sentences were employed as inputs.

The objective of this study is to introduce an automated classification algorithm capable of effectively categorizing short Japanese sentences found in mathematical exercises. To accomplish this, we concentrate on achieving the best agreement between sets of mathematical exercises through the calculation of similarity using weighted word n-gram variance representations. The algorithm is then assessed by comparing it to similar experiments conducted using prediction models, and its accuracy is calculated.

Morphological analysis and relation to reading comprehension

As a study of mathematical morphological analysis, it is popular to investigate the relationship between learners' reading comprehension and their mathematical skills. It is suggested that general vocabulary may serve as a proxy for mathematics-specific vocabulary in studies that do not include measures of mathematics-specific vocabulary (Chow & Ekholm, 2019). Much of the research investigating the relationship between language proficiency and math outcomes focuses specifically on vocabulary for reasons such as memorizing large numbers as words (Spelke & Tsivkin, 2001) and the need to understand oral instruction (Chow & Ekholm, 2019).

While the present study does not specifically address the learners' reading comprehension skills, but we use morphology to analyze the Japanese sentence and to create a vector representation.

Classification of incomplete exercise texts

According to previous research, exercise texts for classification task, which is called "TREC" in the paper, contains the least number of sentences and even the least number of vocabularies of all 7 dataset types, including movie review, sentiment classification dataset and subjectivity dataset (Liu & Guo, 2019). This fact indicates that an exercise text consists of relatively less characters. Previous studies have also shown that it is difficult to achieve adequate performance on the classification of short text by word embedding which was also discussed in Sect. 2.2.2, and therefore another approach is required for this task.

Unlike other natural-language-presented subjects such as languages, history, and social science, mathematical learning materials involve the presentation of notations, formulas, and figures. Using the common PDF format, the processing of non-language information in the mathematical learning materials is costly and complex. Although prior studies have shown that formula processing is detectable if the layout and format are defined (Date & Isozaki, 2015; Fateman et al., 1996), it is difficult to detect when they are not. Such issues arise during the uploading and analyzing of these materials in educational settings due to the challenge of fully extracting content like text, formulas, graphs, and images from published PDFs, leading to incomplete information retrieval (Abekawa & Aizawa, 2016). Hence, other methods should be investigated for the labeling from incomplete text.

In this study, we aim to automatically label the mathematical learning materials by analyzing textual information which is readily extractable from PDF files.

Mathematical education in Japan

Japanese students' performance in mathematics is the highest level among countries in the world, which is said to be due to the influence of students' confidence in mathematics, student Socio-Economic Status (SES), and school emphasis on academic success (Wang et al., 2023).

Japan's Courses of Study are curriculum standards established by the Ministry of Education, Culture, Sports, Science and Technology (MEXT) to ensure that standards are maintained in all schools throughout Japan. They are revised approximately every 10 years. In recent years, the decline in Japan's performance in the PISA 2003 international achievement test has triggered a shift in educational policy toward improving academic achievement (Onishi, 2011). MEXT revision of that standard in 2009 strengthened English foreign language learning and introduced task-based learning (MEXT, 2009). The latest revision, issued in 2018, set three items as learning objectives: "knowledge and skills," "ability to think, judge, express" and "ability to learn and humanity" (MEXT, 2018). Students' textbooks, exercises, and in-class learning are based on the Courses of Study. In mathematics, the curriculum guidelines divide mathematical knowledge and skills into categories, each of which has its own meaning. Table 1 shows the organization of mathematics units and their objectives as defined by the Courses of Study revised in 2009 and exercises in the materials in this study are prepared based on this. Because these standards are used all around Japan, the categorization of exercises can affect mathematical education throughout the country.

Technology is helping researchers better understand how students learn mathematics in order to improve studies on mathematical education (Fishback & Schlicker, 1996; Hussein, 2023). In the context of mathematics in Japan, units on mathematics related to statistics have been introduced at every grade level, as indicated by the enhancement of statistical education, and learning activities using computers and other tools. A recent study has proposed the use of programming environments to support the learning of statistics according to learner's grade (Kayama et al., 2022). To support learners use of such environments, it is important for learners to be able to figure out which exercises are in which grade level of

Table 1 Explanation of each organization part of the units (MEXT, 2009)

Part	Description
I	To provide students with an understanding of numbers and expressions, figures and measurements, quadratic functions, and data analysis, and to cultivate the acquisition of basic knowledge and proficiency in these skills, as well as to cultivate the ability to consider events mathematically, to recognize the merits of mathematics, and to develop an attitude of utilizing these skills
II	To develop understanding of various expressions, figures and equations, exponential and logarithmic functions, trigonometric functions, and differential and integral calculus, to acquire basic knowledge and skills, to develop the ability to consider and express phenomena mathematically, and to foster an attitude to make use of such knowledge
III	To deepen students' understanding of curves on a plane, the complex plane, limits, differential and integral calculus, to develop their knowledge and skills, to develop their ability to consider and express phenomena mathematically, and to foster an attitude to actively utilize these skills
A	To make students understand the number of cases and probability, properties of integers, and properties of figures, to acquire basic knowledge and skills, to cultivate the ability to consider events mathematically, to recognize the goodness of mathematics, and to cultivate an attitude to utilize these skills
B	To develop an understanding of probability distributions and statistical inferences, number sequences and vectors, to acquire basic knowledge and skills, to develop the ability to consider and express events mathematically, and to cultivate an attitude of using these skills

similar statistical units without requiring teacher intervention. Another study has proposed the method to explain the unit structure of textbooks in order to relate knowledge in learning (Taniguchi & Itoh, 2023). However, without knowledge labeling of textbooks and exercises, it is difficult to make use of such unit structures in educational settings.

In this study, we focus on the labeling of mathematics units and verify the assignment of units to textbooks and exercises, which have been the subject of much research. In addition, we focus on the Japanese context of mathematical education and use the most common standards in Japan.

Method

Our goal is to find an algorithm that can assign appropriate labels to educational materials using characters extracted from math teaching material PDFs. In particular, we use the characters extracted from the math teaching material PDFs as input, vectorize them using natural language processing, train the vectors as features, and output the labels l_{pred}.

We defined the method of predicting labels with the following two functions:

$$t \underset{\rightarrow}{f_{vec}} v \underset{\rightarrow}{f_{pred}} l_{pred} \in L \tag{1}$$

where t is a set of characters from a mathematical PDF material, v is a vector from t by vectorization. In the following section, we defined the functions f_{vec_1},f_{vec_2} and f_{pred_1},f_{pred_2} respectively as the methods of vectorization from characters and the methods of prediction from the vector. In other words, we defined f_{vec_1} or f_{vec_2} as the feature-selecting method and did f_{pred_1} or f_{pred_2} as the model-selecting method. Note that there are a set of labels L that l_{pred} can be selected from L. Figure 1 shows the experimental overview from inputting exercise PDF to outputting a prediction.

Data preparation

The input data in this experiment is a Japanese math exercise contained in a PDF file. To use the characters' information of exercises, we first extract text and create a text set from the exercise PDF files. We defined datasets $Q = \{q_1, q_2, ..., q_i, ..., q_n\}$ for each $q_i \in Q$ as an exercises' text data set. Each q has its label $l_i = \{l_{i1}, l_{i2}\}$ in advance, where l_{i1}, l_{i2} represent the 1st level label, and the 2nd level label, respectively. A relation between a unit label and a subunit label can be formulated as follows:

$$\forall_{l_{i_1 2} \in l_{i_1 1}, l_{i_2 2} \in l_{i_2 1}} i_1 1 \neq i_2 1 \Rightarrow i_1 2 \neq i_2 2 \tag{2}$$

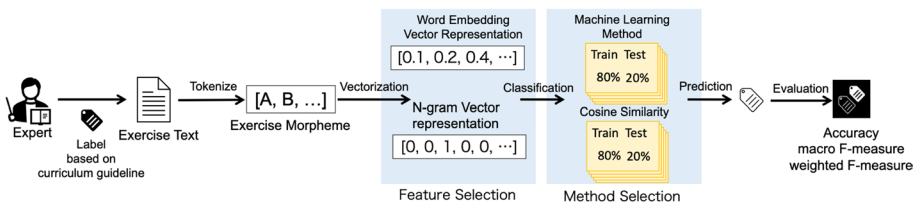

Fig. 1 Overview of experiment

We divide the obtained characters into meaningful chucks before converting them into vectors. This preprocessing provides us with word sets $T_i = \{t_{i_1}, t_{i_2}, ..., t_{i_j}\}$ of each Q_i. $n(T_i)$ equals j where $n(X)$ represents the number of elements in the set X.

The exercise texts used are electronic pdf versions of each of the following exercise books:

- "Supplementary and Revised Edition Charting Mathematics from the Basics I + A"
- "Supplementary and Revised Edition Charting Mathematics from the Basics II + B"
- "Supplementary and Revised Edition Charting Mathematics from the Basics III"
- "Succeeding Mathematics I + A for Textbook Sidelines"
- "Succeeding Mathematics II + B for Textbook Sidelines"
- "Succeeding Mathematics III for Textbook Sidelines"

These exercise books are designed for high school students and align with the textbooks approved by the Japanese government (MEXT, 2021). They are produced by the same company responsible for the widely used textbooks in Japan.

We prepared text files by reading text data using the Python library Pdf2text (Palmer, 2021). Note that PDF files are more difficult to obtain in their complete text form than HTML-formatted files (Ramakrishnan et al., 2012; Smith, 2007).

Japanese high school mathematics teachers created one 1st level unit label and one 2nd level unit label for each exercise by referring to sections in their textbooks and mapping them to each other. There was a total of 2775 exercises, consisting of 24 1st level units and 297 2nd level units. The same 2nd level unit is never assigned across multiple 1st level units. Each 1st level unit consists of between 25 and 200 exercises, with a minimum of five exercises assigned per 2nd level unit. Table 2 shows the content of each 1st level unit, the organization part the unit belongs to, the number of 2nd level units it contains ($n(L_{l2})$), the number of exercises it contains ($n(Q_l)$), and the mean and standard deviation for the morphemes contained in each exercise ($\bar{n(T_l)}, s_{T_l}$). All of these 1st level units are math common standard in Japan. They are categorized into 5 big meaningful sets. The column "part" of Table 2 represents one of the five organization parts (refer to Table 1) that is assigned to the unit. Figure 2 shows an example of the hierarchical structure of 1st level unit and 2nd level unit. Figure 3 shows an example of an exercise and the 1st level unit and 2nd level unit that has been assigned to it.

We used pdf2txt to extract the characters from the mathematical exercise PDF. Figure 4 shows an example of what was extracted from the mathematical exercise PDF. In the figure, (a) represents the raw PDF data of the exercise, (b) represents the extracted Japanese texts from (a), and (c) is an English translation of (b). As shown in (a) of the figure, while the information about the diagram in the PDF cannot be extracted, also the letters highlighted in blue in the PDF do not appear in the extracted text. These words consist of mathematical formulas "GH = 2OG", figures such as "3" of "3点" (3 points), and symbols such as "ABC". It is difficult to extract significant sentences from extracted texts because of the few of the text and symbols related to math equations could be extracted. We can see from (b) or (c) that we could not get the full sentence from PDF text, and it was also somewhat meaningless and difficult to comprehend.

Table 2 Detailed information of each 1st level unit

Unit Name	Part	English translation of the Unit Name	$n(L_{l_2})$	$n(Q_l)$	$\overline{n(T_l)}$	s_{T_l}
数と式	I	Number and formula	18	125	51.8	43.3
集合と命題	I	Set theory	9	60	124.9	79.4
二次関数	I	Quadratic function	15	200	123.3	42.6
図形と計量	I	Measuring graphics	10	85	78.7	49.7
データの分析	I	Data analysis	3	25	107.0	60.2
式と証明	II	Formulas and proofs	10	95	50.3	40.9
複素数と方程式	II	Complex numbers and equations	12	115	66.5	45.1
図形と方程式	II	Geometric equations	16	135	88.1	62.2
三角関数	II	Trigonometric function	11	90	87.7	42.6
指数関数と対数関数	II	Exponential or logarithmic function	8	70	84.1	44.1
微分法	II	Differentiation	6	100	125.0	51.9
積分法	II	Integration	6	70	92.3	41.6
複素数平面	III	Complex plane	8	100	111.7	59.5
極限	III	Limit	12	120	83.2	43.0
式と曲線	III	Equations and curves	18	175	145.2	64.9
微分法の応用	III	High-leveled differentiation	17	175	110.1	46.2
積分法の応用	III	High-leveled integration	18	200	91.6	54.6
場合の数	A	Number of cases	13	120	121.9	104.3
確率	A	Probability	11	90	115.3	74.9
図形の性質	A	Geometric properties	16	110	87.4	54.6
整数の性質	A	Properties of integers	16	160	81.8	61.7
平面上のベクトル	B	Two-dimensional vector	16	125	89.0	52.5
空間のベクトル	B	Three-dimensional vector	10	95	104.4	47.8
数列	B	Sequence	18	135	114.3	57.9
All exercises			297	2775	98.5	61.4

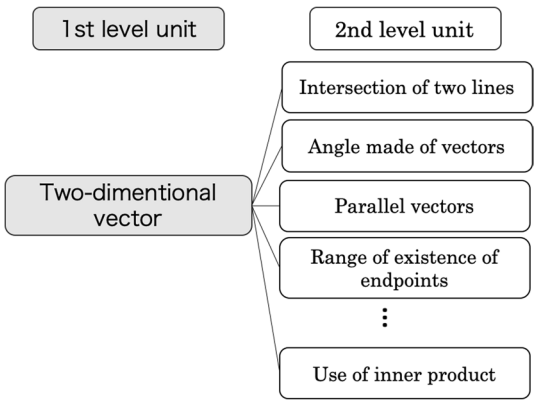

Fig. 2 Example of the hierarchical structure of 1st and 2nd level unit

As Japanese text does not contain word boundaries, preprocessing to extract morphemes is required and we used a package called Nagisa (Ikeda, 2021) for the morphological analysis of text data. Nagisa is a package for the morphological analysis of Japanese sentences. One feature of Nagisa is that it can assign a part of speech to each

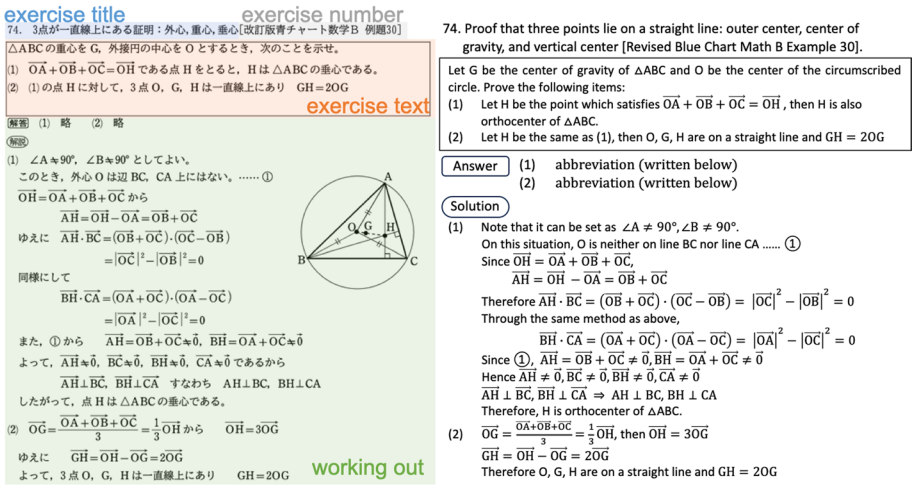

Fig. 3 Example of exercises in the dataset. The 1st level unit "two-dimensional vector" and 2nd level unit "use of inner product" are assigned to an exercise in the figure

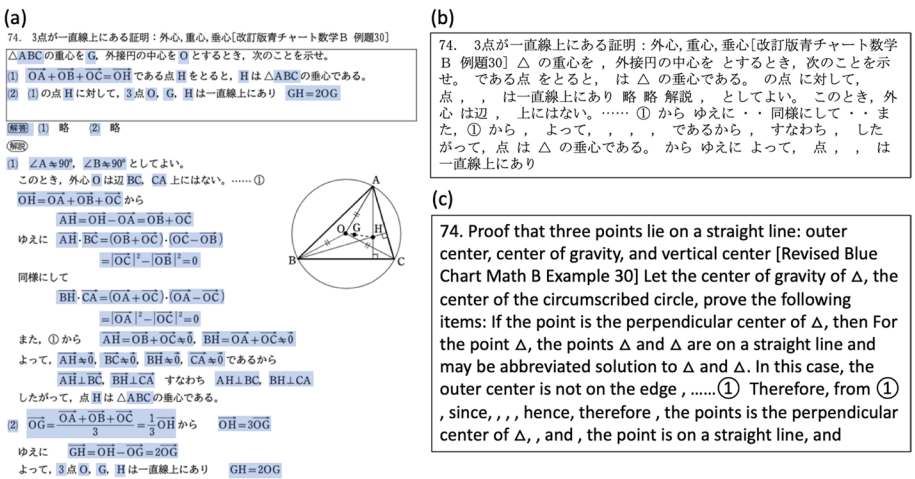

Fig. 4 Examples of PDF and extracted texts

segmented morpheme and can exclude words with a specific part of speech. Some parts of the text cannot precisely be divided into morphemes and therefore some parts of deviation are incorrect.

Vectorization methods

VSM created from N-gram

We assumed that the words or a sequence of words in a sentence which has the same label will be similar to one and another, so we developed the n-gram word extracting method and compared the performance to methods using state-of-the-art word embedding. As we will compare both word embeddings and n-grams in the same context, we have to convert the n-grams into a vector which represents the n-gram features.

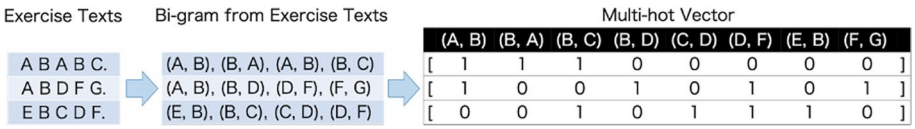

Fig. 5 Example of the way to extract n-grams

We first define vector $V_{G_{i,k}}$ that is created from the specific exercise tokens T_i with all exercise text token T and the number of consecutive tokens of k-gram k, such as:

$$f_{vec_1}(T, T_i, k) \rightarrow V_{G_{i,k}} \tag{3}$$

Figure 5 shows the overview of method to convert n-gram of sentence into vector.

The method of creating n-grams is as follows:

We created a word k-grams $g_{i,k,l}(1 \leq l \leq n(T_i) - k + 1)$ from the tokenized exercise sentences of $t \in T_i$. This means that k consecutive tokens from t_{i_l} to $t_{i_{l+k-1}}$ were taken and stored in a single tuple:

$$g_{i,k,l} = \left(t_{i_l}, t_{i_{l+1}}, \ldots, t_{i_{l+k-1}}\right) \tag{4}$$

Then we made $G_{i,k}$ aggregating all l of $g_{i,k,l}$.

$$G_{i,k} = g_{i,k,1}, g_{i,k,2}, \ldots, g_{i,k,n(T_i)-k+1} \tag{5}$$

For vectorization using word n-grams, we prepared a list G_k that includes all $g_{i,k,l}$ in all $G_{i,k}$. Then, we made a list called k-gram-list that indicates if each component of the n -grams included the query n-grams. We defined the mth elements of k-gram-list:

$$G_k = \left\{g_{x,k,z} | \exists_{x,z}(x = i) \wedge (z = l)\right\} \tag{6}$$

$$k - gram - list[m] = g_m\left(g_m \in G_k, \forall_{x,y}, x \neq y \Rightarrow g_x \neq g_y, 0 \leq m < n(G_k)\right) \tag{7}$$

For each i, the q_i should have one vector whose length is the same as $n(G_k)$. The ith value of v at k-gram, $V_{G_{i,k},m} \in V_{G_{i,k}}$, is determined by the following formula:

$$V_{G_{i,k},m} = \begin{cases} 1\left(g_m \in G_{i,k}\right) \\ 0(\text{otherwise}) \end{cases} \tag{8}$$

When Nagisa morphologically analyzes numbers, it recognizes each number as a one-digit noun. In mathematical texts, different numerals are treated as different morphemes, so we created an algorithm that treats digits as a single number, as shown in

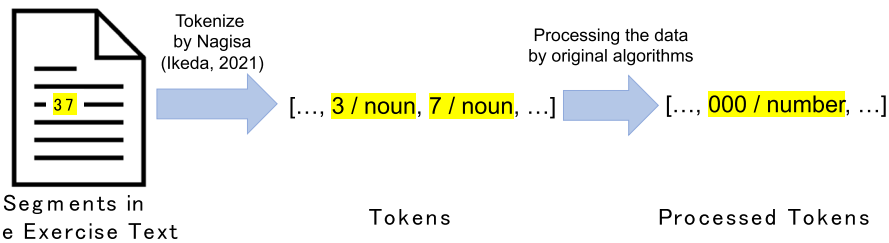

Fig. 6 Example formula processing added to Nagisa

Table 3 The number of n-gram elements in the vectors

n	**Number of n-gram elements**
1	2451
2	20,424
3	56,324
4	95,536
5	130,034
6	157,546

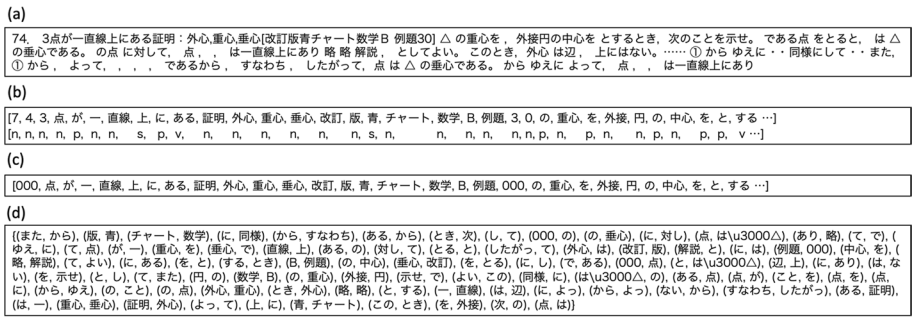

Fig. 7 Processing flow of creating n-grams with Nagisa

Fig. 6, and treated all numbers as the same thing. This process makes easier to find the same exercise except for numbers or formula.

We collected the n-gram data of the exercise texts. In n-grams, it is necessary to determine the value of n for good classification accuracy. Although there are some studies that explore appropriate values of n for each task, as research has shown that large n-grams have advantages in generating features that can be interpreted in malware analysis (Raff et al., 2018), in almost all previous studies n values are very small, and $n > 6$ is extremely rare. Larger values of n are not tested due to the computational burden and the risk of overfitting. So in this study, we conducted n-g extraction for $1 \leq n \leq 6$. Table 3 shows the results of the number of n-grams with $1 \leq n \leq 6$. Figure 7 shows the overall flow of creating n-grams with Nagisa. In the figure, (a) represents the extracted full text data. The item (b) represents a list of morphemes with part of the speech of each morpheme: n, p, v, s stands for noun, particle, verb, suffix, respectively. The item (c) represents an obtained list of morphemes processed numbers by the method illustrated in Fig. 6. The item (d) represents the completely obtained bi-gram from (a).

Word embedding vectorization We defined vector V_{E_i} that is created from the specific exercise tokens $T_i\,(1 \leq i \leq n(Q))$ and the model for word embedding model, i.e.

$$f_{vec_2}(T_i, model) \rightarrow V_{E_i} \tag{9}$$

For vectorization with word embedding, we used a model called fastText (Joulin et al., 2017). There is a website, https://fasttext.cc/docs/en/crawl-vectors.html, which has pre-trained models for 157 languages. In this experiment, we used the Japanese model,

which combines three methods to represent input sentence data in 300 dimensions: character 5-gram, weighting by position, and Word2Vec (Church, 2017).

Label prediction by vectorized sentence

Prediction by calculating cosine similarity

For any exercises text T, we use score $s(T_a, T_b)$ to measure the similarity of texts between T_a and T_b. The higher $s(T_a, T_b)$ is, the more similar T_a and T_b are. The answer of predicting labels with finding similarity of exercises can be formulated as: Given a set of query exercise text vector v_{query}, a labeled-exercise text $v_{labeled}$ that has the label $l_{labeled}$, weight parameters function f_w, our goal is to integrate these heterogeneous materials to measure the similarity scores of exercise pairs and predict the 1st level unit label or the 2nd level unit label for any v_{query} by selecting the candidate label l_{pred} with a predicted label, i.e.

$$f_{pred_1}\left(v_{query}, V_{labeled}, L_{labeled}, f_w\right) \rightarrow l_{pred} \in L \tag{10}$$

where $V_{labeled}, L_{labeled}$ is the set of vectors of labeled data and labels of them respectively, f_w is the weight parameters function, and L is the domain of labels in the data. The selected label for query l_{pred} is the prediction label of the exercises.

In this algorithm, as shown in Fig. 1, the data set is divided into label data and query, and the similarity between the set of word n-grams in the label data and the set of word n-grams in the query is calculated. Here, the similarity of the vectors is the values$\left(v_{l_X}, v_{query}\right)$ obtained using the cosine similarity method, where v_{l_X}, v_{query} represent the vector of word n-grams of the labeled data with the label $l_X (1 \leq X \leq n(L))$ and the vector of word n-grams of the query, respectively.

$$s\left(v_{l_X}, v_{query}\right) = \frac{v_{l_X} \cdot v_{query}}{\|v_{l_X}\| \|v_{query}\|} \tag{11}$$

We then compute $s_{l_X,query}$ by aggregating $s\left(v_{l_X}, v_{query}\right)$ of all vector v_{l_X} with label l_X, and substitute them all into the determined weight function f_w. Previous studies improve accuracy by weighting for realistic non-homogeneous data sets. One study successfully achieved high accuracy using cosine similarity with added weighting to effectively train CNNs in realistic learning situations such as class imbalance, small size, and label noise (Kobayashi, 2021). Weighting explanatory variables with generated n-grams is said to be an effective means of improving text classification accuracy (Graovac et al., 2015). The calculation formula of $s_{l_X,query}$ is as follows:

$$s_{l_X,query} = f_w\left(s_{set_{l_X,query}}\right) \tag{12}$$

$$s_{set_{l_X,query}} = \left\{ s\left(v_{l_{X_1}}, v_{query}\right), s\left(v_{l_{X_2}}, v_{query}\right), \ldots, s\left(v_{l_{X_{n\left(V_{labeled_X}\right)}}}, v_{query}\right) \right\} \tag{13}$$

where $V_{labeled_X}$ represents the labeled vector assigned label X. Finally, we find $s_{l_X,query}$ for all l_X and determine $l_{pred,query}$ as follows:

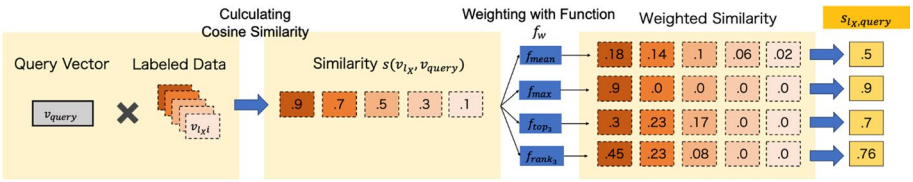

Fig. 8 How to find the weight at which label l_X assigns to a query

$$l_{pred,query} = \underset{l_X(X=1,2,\ldots,n(L))}{\operatorname{argmax}} s_{l_X,query} \tag{14}$$

What this formula means is that the predicted label is the same label as the problem with higher similarity. Various changes in the function f_w are used to determine a more suitable weighting for classification. In this experiment, we defined the functions as follows:

$$f_{mean}\left(s_{set_{l_X,query}}\right) = \sum_{k=1}^{n\left(s_{set_{l_X,query}}\right)} \frac{s_{l_{X_k},query}}{n\left(s_{set_{l_X,query}}\right)} \tag{15}$$

$$f_{max}\left(s_{set_{l_X,query}}\right) = H_{X,1} \tag{16}$$

$$f_{top_m}\left(s_{set_{l_X,query}}\right) = \sum_{k=1}^{m} \frac{H_{X,k}}{m} \tag{17}$$

$$f_{rank_m}\left(s_{set_{l_X,query}}\right) = \sum_{k=1}^{m} \frac{(m-k+1)H_{X,k}}{\sum_{k=1}^{m} k} \tag{18}$$

where $H_{X,k}$ represents the kth highest value in $s_{set_{l_X},query}$. The prediction vector for query, v_{query}, is defined as the array of values $[s_{l_1,query}, s_{l_2,query}, \ldots, s_{l_{n(L)},query}]$ obtained by the function f_w.

We created these functions based on the sentence similarity which has the same label: the more similar the sentences are, the more likely to have the same label. There are two assumptions as follows:

- Assumption 1: Any pair of two exercises that have the same label are similar to each other. Therefore, we created to find the most appropriate label considering all labeled exercises' similarities, f_{mean}.
- Assumption 2: Specific exercises with the same label have high similarity with each other. Therefore, we created to find the most appropriate label considering m labeled exercises' similarities, f_{top_m},f_{rank_m}.

Figure 8 shows the overview of how to find the weight of specific label assigns to a query.

Prediction by machine learning

The problem of finding similar exercises can be formulated as follows: Given a set of test exercise text vector v_{test}, a set of training text vectors v_{train} that have the true label set, our goal is to integrate these heterogeneous materials to predict the 1st level unit or the 2nd level unit for any vector from query exercise text v_{query} by selecting the candidate label l_{pred} with a predicted label, i.e.

$$f_{pred_2}(v_{test}, v_{train}, model) \rightarrow l_{pred} \in L \tag{19}$$

where model is a package that can classify these vectors into the specific number of categories and L is the domain of labels in the data. The selected label for test data v_{test} is the prediction label of the exercises, described as l_{pred}.

- XGBoost (Chen et al., 2015; Chen & Guestrin, 2016): This model, which merges boosting with decision trees, has demonstrated promising outcomes in diverse natural language processing assignments, making it an appropriate choice for employment in this paper's context.
- Random Forest (Breiman, 2001): This is a model that employs numerous decision trees trained using randomly selected training data. It performs effectively even with a considerable number of explanatory variables, enabling it to handle a 300-dimensional vector.
- Logistic Regression (Cox, 1958), Perceptron (Rosenblatt, 1958): Both models are used for statistical regression with variables that follow a Bernoulli distribution. However, the former employs coordinate descent or quasi-Newtonian methods for parameter determination in optimization problems, whereas the latter utilizes the stochastic gradient descent method.

Evaluation

We conducted experiments using fivefold cross validation for training and prediction. The use of fivefold reduces over-training on training and label data. In addition, accuracy A_L, macro F-measure F_L and weighted F-measure F_{wL} were used to evaluate this experimental algorithm. Let TP_l, FP_l, TN_l and FN_l denote that the true prediction for a label l is correct or wrong, and that the false prediction for a label l is correct or wrong, then accuracy A_l and precision P_l, recall R_l and the f score F_l can be expressed as follows.

$$P_l = \frac{TP_l}{TP_l+FP_l}, R_l = \frac{TP_l}{TP_l+FN_l}, A_l = \frac{TP_l+TN_l}{TP_l+TN_l+FP_l+FN_l}, F_l = \frac{2P_lR_l}{P_l+R_l} \tag{20}$$

$$P_L = \frac{\sum_{l\in L}P_l}{n(L)}, R_L = \frac{\sum_{l\in L}R_l}{n(L)}, A_L = \frac{\sum_{l\in L}A_l}{n(L)}, F_L = \frac{\sum_{l\in L}F_l}{n(L)}, F_{wL} = \frac{2P_LR_L}{P_L+R_L} \tag{21}$$

We used A_L, F_L, and F_{wL} to evaluate the performance of the prediction.

Result

Classification results with selecting features and methods

We take n-grams of $1 \leq n \leq 6$ and vector with w2vec into consideration. We also prepare a cosine similarity model with the weighted function formula (15), (16), (17), (18)

Table 4 Classification in 1st level unit between a feature and accuracy A_L in n-grams and machine learning methods

	$n=1$	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$	**w2vec**
Similarity (f_{mean})	0.7121	0.7398	0.7059	0.7049	0.6825	0.6858	0.3834
Similarity (f_{max})	0.8025	0.8184	0.8130	0.7953	0.7791	0.7539	0.6944
Similarity (f_{top_2})	0.8328	0.8386	0.8314	0.8105	0.7838	0.7647	0.7063
Similarity (f_{top_3})	0.8411	0.8429	0.8357	0.8072	0.7791	0.7625	0.7081
Similarity (f_{top_4})	0.8407	0.8494	0.8328	0.8105	0.7730	0.7539	0.7049
Similarity (f_{top_5})	0.8432	0.8505	0.8324	0.8022	0.7679	0.7506	0.6955
Similarity (f_{top_6})	0.8443	0.8494	0.8321	0.7960	0.7607	0.7409	0.6818
Similarity (f_{top_7})	0.8404	0.8523	0.8303	0.7888	0.7596	0.7373	0.6728
Similarity (f_{top_8})	0.8389	0.8508	0.8256	0.7859	0.7553	0.7348	0.6595
Similarity (f_{top_9})	0.8378	0.8483	0.8202	0.7831	0.7524	0.7243	0.6541
Similarity ($f_{top_{10}}$)	0.8350	0.8465	0.8162	0.7798	0.7481	0.7196	0.6436
Similarity (f_{rank_2})	0.8231	0.8382	0.8281	0.8090	0.7841	0.7647	0.7095
Similarity (f_{rank_3})	0.8353	0.8411	0.8332	0.8123	0.7852	0.7665	0.7114
Similarity (f_{rank_4})	0.8400	0.8472	0.8364	0.8130	0.7852	0.7672	0.7106
Similarity (f_{rank_5})	0.8432	0.8523	0.8382	0.8119	0.7813	0.7654	0.7085
Similarity (f_{rank_6})	0.8461	0.8533	0.8371	0.8141	0.7798	0.7589	0.7067
Similarity (f_{rank_7})	0.8458	0.8537	0.8386	0.8123	0.7780	0.7593	0.6998
Similarity (f_{rank_8})	0.8483	0.8551	0.8375	0.8101	0.7744	0.7557	0.6951
Similarity (f_{rank_9})	0.8479	0.8555	0.8353	0.8072	0.7759	0.7517	0.6923
Similarity ($f_{rank_{10}}$)	0.8479	0.8537	0.8335	0.8029	0.7719	0.7463	0.6836
xgb	0.8605	0.8274	0.6681	0.5831	0.4825	0.4263	0.6468
rf	0.9250	**0.9013**	0.8310	0.7427	0.6526	0.5968	0.7139
mlp	0.7910	0.8732	0.8631	**0.8674**	**0.8382**	**0.8209**	0.8065
lr	0.9193	0.8930	**0.8641**	0.8371	0.8025	0.7492	**0.8537**

($2 \leq m \leq 10$), and machine learning method Xgboost, Random Forest, Perceptron and Logistic Regression. Tables 4, 5, 6, 7, 8 and 9 show the three kinds of prediction result, accuracy A_L, macro F-measure F_L and weighted F-measure F_{wL}, when we used the combination of each feature and the model. In the tables, the best performance rate in each feature is bolded, and the best overall performance is underlined. We also draw a graph that represents all recalls of each feature and model selection in Figs. 9 and 10. The tables show that at the both 1st level unit and 2nd level unit prediction, the algorithm yielded the best A_L, F_L of all when using mono-gram features with the Random Forest model, and best F_{wL} when bi-gram features were used with the Random Forest model, when compared to the use of word embedding, n-grams of the other n features, and the other models such as cosine similarity or machine learning methods.

Unlike word embedding, n-grams can be analyzed literally without considering the context. It is a suitable feature for this experiment in that we are using text data poorly extracted from PDF files. In addition, since the experiment using cosine similarity considers textual similarity, the text is likely to be classified into the units that contain many texts with high similarity. Therefore, the higher the textual similarity of the texts, the higher the similarity at a larger n is likely to be. However, if n is too large, there will be fewer matching n-gram words and less textual similarity. Considering these conditions,

Table 5 Classification in 1st level unit between a feature and macro F-measure F_L in n-grams and machine learning methods

	$n=1$	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$	**w2vec**
Similarity (f_{mean})	0.7211	0.7479	0.7122	0.7058	0.6827	0.6835	0.3894
Similarity (f_{max})	0.8015	0.8124	0.8051	0.7864	0.7684	0.7437	0.7031
Similarity (f_{top_2})	0.8309	0.8345	0.8236	0.8004	0.7746	0.7550	0.7133
Similarity (f_{top_3})	0.8369	0.8375	0.8282	0.7969	0.7695	0.7520	0.7133
Similarity (f_{top_4})	0.8370	0.8441	0.8249	0.8007	0.7607	0.7427	0.7094
Similarity (f_{top_5})	0.8392	0.8457	0.8252	0.7928	0.7542	0.7369	0.7014
Similarity (f_{top_6})	0.8394	0.8444	0.8243	0.7836	0.7444	0.7257	0.6868
Similarity (f_{top_7})	0.8338	0.8464	0.8212	0.7728	0.7438	0.7199	0.6767
Similarity (f_{top_8})	0.8328	0.8457	0.8160	0.7699	0.7389	0.7154	0.6619
Similarity (f_{top_9})	0.8312	0.8421	0.8091	0.7669	0.7330	0.7004	0.6563
Similarity ($f_{top_{10}}$)	0.8286	0.8387	0.8013	0.7598	0.7226	0.6913	0.6440
Similarity (f_{rank_2})	0.8209	0.8343	0.8196	0.7999	0.7740	0.7546	0.7163
Similarity (f_{rank_3})	0.8330	0.8379	0.8254	0.8018	0.7756	0.7572	0.7181
Similarity (f_{rank_4})	0.8379	0.8433	0.8288	0.8027	0.7752	0.7570	0.7162
Similarity (f_{rank_5})	0.8406	0.8484	0.8304	0.8020	0.7708	0.7552	0.7139
Similarity (f_{rank_6})	0.8433	0.8485	0.8297	0.8046	0.7666	0.7472	0.7119
Similarity (f_{rank_7})	0.8424	0.8492	0.8310	0.8028	0.7649	0.7473	0.7050
Similarity (f_{rank_8})	0.8439	0.8502	0.8298	0.7995	0.7605	0.7421	0.7002
Similarity (f_{rank_9})	0.8418	0.8504	0.8275	0.7967	0.7586	0.7375	0.6965
Similarity ($f_{rank_{10}}$)	0.8414	0.8490	0.8251	0.7904	0.7551	0.7303	0.6865
xgb	0.8572	0.8212	0.6171	0.5195	0.3938	0.3312	0.6303
rf	0.9250	**0.8996**	0.8205	0.7133	0.6176	0.5568	0.7098
mlp	0.7914	0.8719	**0.8623**	**0.8665**	**0.8393**	**0.8178**	0.8079
lr	0.9218	0.8903	0.8570	0.8247	0.7880	0.7358	**0.8566**

mono-grams turn out to be the most suitable n-size since it is the size of the n-gram that is most likely to be used in the experiment.

Figures 11 and 12 compare the graph of A_L, F_L, F_{wL} between selected weighted similarity models and MLP models. The reason of selecting the model in the figure is clarified as follows: f_{top_3} is the best prediction model of all f_{top_m} models, f_{rank_3} is the best prediction model of all f_{rank_m} models in 2nd level unit prediction, and f_{rank_9} is the best prediction model of all f_{rank_m} models in 1st level unit prediction.

As shown in Figs. 11 and 12, in both experiments, we could see the results using weighted similarity models are similar to that using MLP models from the point of the shape of the figure, while the result between MLP and the other machine learning methods' results are not so similar; the former doesn't have a peak when $n=1$, and the latter does when $n=1$. This suggests that weighted similarity models are taking the same method as MLP, like aggregating the number in the way of calculating the prediction. This also shows that as for the optimal value of n for n-grams, $n=2$ was optimal for prediction by searching for similar sentences using cosine similarity. This means that the smaller the value of n, the greater the number of matching components, while the larger the value of n, the higher the degree of similarity of sentences with the higher agreement, suggesting that $n=2$ is a moderate value that covers both aspects.

Table 6 Classification in 1st level unit between a feature and weighted F-measure F_{wL} in n-grams and machine learning methods

	$n=1$	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$	**w2vec**
Similarity (f_{mean})	0.7127	0.7392	0.7055	0.7063	0.6831	0.6864	0.3962
Similarity (f_{max})	0.8024	0.8191	0.8140	0.7960	0.7797	0.7541	0.6939
Similarity (f_{top_2})	0.8331	0.8393	0.8322	0.8109	0.7837	0.7638	0.7064
Similarity (f_{top_3})	0.8414	0.8440	0.8365	0.8073	0.7785	0.7609	0.7091
Similarity (f_{top_4})	0.8408	0.8505	0.8337	0.8105	0.7719	0.7514	0.7059
Similarity (f_{top_5})	0.8434	0.8514	0.8332	0.8019	0.7659	0.7476	0.6961
Similarity (f_{top_6})	0.8446	0.8502	0.8331	0.7954	0.7584	0.7378	0.6825
Similarity (f_{top_7})	0.8407	0.8531	0.8313	0.7886	0.7569	0.7344	0.6735
Similarity (f_{top_8})	0.8394	0.8515	0.8265	0.7852	0.7528	0.7321	0.6607
Similarity (f_{top_9})	0.8385	0.8492	0.8212	0.7823	0.7504	0.7223	0.6556
Similarity ($f_{top_{10}}$)	0.8358	0.8475	0.8175	0.7795	0.7466	0.7184	0.6457
Similarity (f_{rank_2})	0.8232	0.8389	0.8291	0.8093	0.7844	0.7643	0.7097
Similarity (f_{rank_3})	0.8357	0.8419	0.8340	0.8126	0.7850	0.7656	0.7118
Similarity (f_{rank_4})	0.8403	0.8481	0.8372	0.8134	0.7846	0.7659	0.7116
Similarity (f_{rank_5})	0.8434	0.8532	0.8391	0.8121	0.7807	0.7638	0.7094
Similarity (f_{rank_6})	0.8462	0.8544	0.8377	0.8139	0.7792	0.7568	0.7076
Similarity (f_{rank_7})	0.8460	0.8547	0.8392	0.8121	0.7767	0.7566	0.7006
Similarity (f_{rank_8})	0.8484	0.8562	0.8383	0.8099	0.7726	0.7527	0.6959
Similarity (f_{rank_9})	0.8483	0.8563	0.8364	0.8069	0.7739	0.7486	0.6929
Similarity ($f_{rank_{10}}$)	0.8484	0.8545	0.8344	0.8025	0.7698	0.7431	0.6846
xgb	0.8601	0.8279	0.6777	0.6037	0.5274	0.4800	0.6530
rf	0.9255	**0.9020**	0.8315	0.7430	0.6514	0.6016	0.7191
mlp	0.7924	0.8745	**0.8636**	**0.8673**	**0.8367**	**0.8192**	0.8066
lr	0.9193	0.8935	0.8655	0.8387	0.8029	0.7453	**0.8537**

Random forest mono-gram feature analysis

To examine the predictions in detail, we performed feature analysis on the random forest model that was trained using monograms as it had the highest accuracy of all of the models that were evaluated. Table 10 (a) contains the most influential monograms and their degree of influence. The words 'I', 'III', 'II', 'A', and 'B' appear to be highly influential. This is because, as shown in the figure, the classifications of the units fall into one of these five patterns. Therefore, when these classifications are listed in the PDFs, it was found that these words can be used to classify the unit more.

Also, not all PDFs contain a classification indicating these five categories. The word "解説" (solution) is not a word that describes the math exercises or the solutions themselves. Therefore, by omitting these as stop-words, shaded in gray in Table 10 (a), the prediction can be performed to obtain a more general classification prediction result. This prediction resulted in A_L of 82.88%, F_L of 82.82%, and F_{wL} of 83.08%. Table 10 (b) shows the most influential words and their degree of influence in this prediction. The top five words were words representing "ベクトル" (vector), "数" (number), "関数" (function), "確率" (probability) and "複素" (complex) respectively. All of these words are used as part of more than one name of a specific unit. Therefore, it is likely that these words were helpful in classifying the text into broad categories. Note that the assertion of the organization part in a specific place in the

Table 7 Classification in 2nd level unit between a feature and accuracy A_L in n-grams and machine learning methods

	$n=1$	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$	**w2vec**
Similarity (f_{mean})	0.4317	0.4699	0.4447	0.4418	0.4249	0.4151	0.2584
Similarity (f_{max})	0.5589	0.6076	0.5946	0.5780	0.5564	0.5305	0.5121
Similarity (f_{top_2})	0.5845	0.6252	0.6050	0.5914	0.5640	0.5337	0.4941
Similarity (f_{top_3})	0.5795	0.6169	0.6000	0.5823	0.5514	0.5283	0.4688
Similarity (f_{top_4})	0.5553	0.5968	0.5845	0.5607	0.5359	0.5164	0.4328
Similarity (f_{top_5})	0.4840	0.5586	0.5510	0.5431	0.5232	0.5074	0.3640
Similarity (f_{top_6})	0.4663	0.5276	0.5276	0.5222	0.5085	0.4984	0.3495
Similarity (f_{top_7})	0.4544	0.5016	0.5099	0.5045	0.4905	0.4890	0.3240
Similarity (f_{top_8})	0.4411	0.4829	0.4908	0.4915	0.4861	0.4836	0.2998
Similarity (f_{top_9})	0.3795	0.4490	0.4688	0.4782	0.4764	0.4782	0.2378
Similarity ($f_{top_{10}}$)	0.3467	0.4231	0.4468	0.4631	0.4681	0.4710	0.2288
Similarity (f_{rank_2})	0.5895	0.6335	0.6083	0.5924	0.5676	0.5434	0.5114
Similarity (f_{rank_3})	0.5917	0.6364	0.6119	0.5993	0.5679	0.5409	0.4915
Similarity (f_{rank_4})	0.5813	0.6299	0.6054	0.5921	0.5604	0.5402	0.4742
Similarity (f_{rank_5})	0.5514	0.6119	0.5989	0.5809	0.5557	0.5359	0.3968
Similarity (f_{rank_6})	0.5124	0.5859	0.5841	0.5705	0.5467	0.5319	0.3838
Similarity (f_{rank_7})	0.4923	0.5712	0.5723	0.5640	0.5416	0.5250	0.3726
Similarity (f_{rank_8})	0.4782	0.5532	0.5582	0.5553	0.5337	0.5182	0.3582
Similarity (f_{rank_9})	0.4584	0.5377	0.5467	0.5409	0.5279	0.5114	0.2847
Similarity ($f_{rank_{10}}$)	0.4321	0.5178	0.5315	0.5330	0.5214	0.5095	0.2656
xgb	0.1441	0.0868	0.0659	0.0541	0.0490	0.0375	0.1643
rf	0.6850	**0.6829**	0.6314	0.5957	0.5452	0.4861	0.4987
mlp	0.4418	0.6281	**0.6544**	**0.6566**	**0.6414**	**0.6169**	0.4840
lr	0.6404	0.6339	0.6270	0.6220	0.5924	0.5636	**0.6245**

PDF would be helpful in classifying exercises, however less generalizable as it would rely on a consistent format that might not be realistic.

Discussion

Feature selection of extracted incomplete text from PDFs

Labeling incomplete text has been tackled in previous research by using n-grams, which was shown to be an effective way to meet this problem (Cavnar & Trenkle, 1994; Graovac, 2014; Suen, 1979). In the present research, we investigated using n-grams on the extracted texts from a PDF of mathematical exercises for which complete texts were difficult to obtain and categorized them into different leveled units. First, the extracted text could not pick up any information such as mathematical equations, symbols, or numbers. When predicting the topic of incomplete texts, we found that vector classification, which involves only information on whether the text is composed of similar elements and does not involve contextual analysis such as n-grams, was more effective than models that involve contextual analysis. However, we found that mono-grams which are similar to more traditional methods, such as n-grams or bag of words, provided the best classification performance, contradicting results from previous research for this specific task. Therefore, we assume that the use of n-grams in the classification of incomplete texts may depend on the target of the task, which

Table 8 Classification in 2nd level unit between a feature and macro F-measure F_L in n-grams and machine learning methods

	$n=1$	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$	**w2vec**
Similarity (f_{mean})	0.4046	0.4461	0.4288	0.4319	0.4190	0.4094	0.2459
Similarity (f_{max})	0.5103	0.5573	0.5411	0.5153	0.4958	0.4723	0.4817
Similarity (f_{top_2})	0.5167	0.5616	0.5380	0.5216	0.4979	0.4725	0.4388
Similarity (f_{top_3})	0.4903	0.5380	0.5182	0.4974	0.4726	0.4589	0.3904
Similarity (f_{top_4})	0.4419	0.4933	0.4832	0.4666	0.4511	0.4395	0.3307
Similarity (f_{top_5})	0.2694	0.4082	0.4209	0.4323	0.4297	0.4263	0.1895
Similarity (f_{top_6})	0.2431	0.3376	0.3731	0.3968	0.4038	0.4150	0.1797
Similarity (f_{top_7})	0.2339	0.2905	0.3413	0.3675	0.3799	0.3994	0.1592
Similarity (f_{top_8})	0.2215	0.2649	0.3110	0.3485	0.3687	0.3927	0.1388
Similarity (f_{top_9})	0.1604	0.2310	0.2883	0.3283	0.3548	0.3847	0.0740
Similarity ($f_{top_{10}}$)	0.1247	0.2061	0.2619	0.3061	0.3434	0.3789	0.0704
Similarity (f_{rank_2})	0.5275	0.5769	0.5462	0.5274	0.5018	0.4826	0.4661
Similarity (f_{rank_3})	0.5202	0.5703	0.5422	0.5273	0.5006	0.4772	0.4291
Similarity (f_{rank_4})	0.4914	0.5510	0.5253	0.5121	0.4828	0.4725	0.3919
Similarity (f_{rank_5})	0.4177	0.5179	0.5063	0.4917	0.4770	0.4620	0.2148
Similarity (f_{rank_6})	0.3283	0.4582	0.4770	0.4748	0.4634	0.4565	0.2050
Similarity (f_{rank_7})	0.2814	0.4250	0.4551	0.4640	0.4522	0.4476	0.1958
Similarity (f_{rank_8})	0.2552	0.3924	0.4267	0.4488	0.4414	0.4391	0.1829
Similarity (f_{rank_9})	0.2367	0.3609	0.4107	0.4259	0.4349	0.4317	0.1047
Similarity ($f_{rank_{10}}$)	0.2113	0.3284	0.3878	0.4150	0.4247	0.4280	0.0862
xgb	0.0333	0.0150	0.0087	0.0036	0.0026	0.0014	0.0480
rf	0.6128	**0.5967**	0.5478	0.5151	0.4623	0.4128	0.4285
mlp	0.3919	0.5725	**0.5999**	**0.6084**	**0.5950**	**0.5707**	0.4377
lr	0.5858	0.5828	0.5752	0.5745	0.5499	0.5385	**0.5922**

in this case was Japanese mathematical exercises. As the previous research that successfully utilized n-grams to classify incomplete text (Cavnar & Trenkle, 1994; Graovac, 2014; Suen, 1979) neither targeted Japanese nor mathematical exercises, this may have implications for future research into the classification of incomplete Japanese or mathematical texts.

Model selection for more precise prediction

We aimed at labeling Japanese math text more precisely. A previous study treating Japanese mathematical exercises' text classification yields 79.57% accuracy with WE-KE model (Tian et al., 2022). In this experiment, proposed algorithms predicted different leveled units by two methods: search by similarity sentences using cosine similarity and classification using machine learning. The results concluded that the best prediction accuracy was achieved using Random Forest, which resulted in 92.50%. This shows that our method using mono-grams and Random Forest performed well when it comes to Japanese mathematical text classification.

The result indicates that mono-grams yielded the best classification when we used the method of Random Forest classification. The reason that mono-gram performs well is, as Fig. 3 shows, there are incomplete parts of the text when extracting from PDF files, so there is some meaningless parts that consist of multiple words within a chunk. In

Table 9 Classification in 2nd level unit between a feature and weighted F-measure F_{wL} in n-grams and machine learning methods

	$n=1$	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$	**w2vec**
Similarity (f_{mean})	0.4468	0.4799	0.4495	0.4388	0.4159	0.4051	0.2879
Similarity (f_{max})	0.5641	0.6140	0.6021	0.5860	0.5617	0.5349	0.5183
Similarity (f_{top_2})	0.6025	0.6419	0.6229	0.6071	0.5768	0.5439	0.5136
Similarity (f_{top_3})	0.6084	0.6422	0.6249	0.6036	0.5675	0.5406	0.5016
Similarity (f_{top_4})	0.5966	0.6330	0.6186	0.5883	0.5563	0.5330	0.4755
Similarity (f_{top_5})	0.5605	0.6115	0.5949	0.5768	0.5472	0.5259	0.4319
Similarity (f_{top_6})	0.5478	0.5961	0.5812	0.5610	0.5367	0.5181	0.4178
Similarity (f_{top_7})	0.5368	0.5788	0.5707	0.5496	0.5218	0.5118	0.3932
Similarity (f_{top_8})	0.5265	0.5653	0.5565	0.5401	0.5209	0.5081	0.3711
Similarity (f_{top_9})	0.4780	0.5368	0.5374	0.5296	0.5129	0.5036	0.3232
Similarity ($f_{top_{10}}$)	0.4551	0.5149	0.5184	0.5174	0.5065	0.4963	0.3119
Similarity (f_{rank_2})	0.6045	0.6470	0.6238	0.6062	0.5800	0.5544	0.5261
Similarity (f_{rank_3})	0.6131	0.6561	0.6319	0.6171	0.5821	0.5519	0.5152
Similarity (f_{rank_4})	0.6116	0.6556	0.6303	0.6134	0.5772	0.5530	0.5075
Similarity (f_{rank_5})	0.5999	0.6454	0.6293	0.6062	0.5730	0.5519	0.4640
Similarity (f_{rank_6})	0.5782	0.6311	0.6211	0.5985	0.5674	0.5488	0.4511
Similarity (f_{rank_7})	0.5672	0.6232	0.6131	0.5938	0.5645	0.5432	0.4410
Similarity (f_{rank_8})	0.5581	0.6102	0.6043	0.5886	0.5584	0.5376	0.4281
Similarity (f_{rank_9})	0.5412	0.6024	0.5950	0.5781	0.5528	0.5310	0.3735
Similarity ($f_{rank_{10}}$)	0.5203	0.5886	0.5839	0.5716	0.5485	0.5294	0.3570
xgb	0.2162	0.1378	0.1105	0.0971	0.0895	0.0700	0.2308
rf	**0.7064**	0.7099	0.6592	0.6213	0.5660	0.4969	0.5260
mlp	0.4479	0.6461	**0.6692**	**0.6702**	**0.6533**	**0.6280**	0.4867
lr	0.6533	0.6397	0.6349	0.6305	0.5939	0.5525	**0.6311**

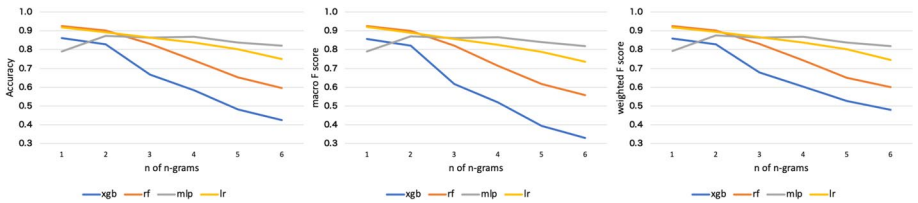

Fig. 9 Relationship between n value in n-gram and evaluation values with some machine learning models in 1st level unit prediction

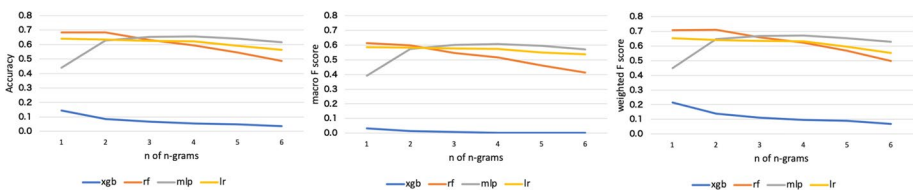

Fig. 10 Relationship between n value in n-gram and evaluation values with some machine learning models in 2nd level unit prediction

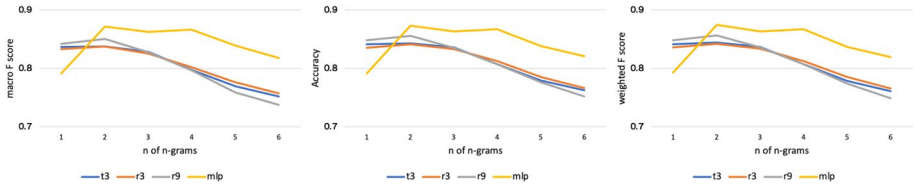

Fig. 11 Relationship between n value in n-gram and evaluation values with MLP and aggregate function methods in 1st level unit prediction

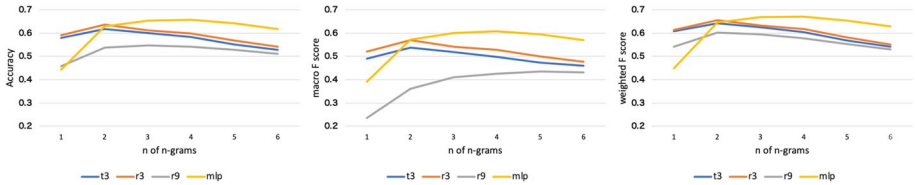

Fig. 12 Relationship between n value in n-gram and evaluation values with MLP and aggregate function methods in 2nd level unit prediction

Table 10 Feature analysis in mono-gram and Random Forest

(a) without omitting any words			(b) omitting meaningless words		
Mono-gram	**Mono-gram (English)**	**Importance**	**Mono-gram**	**Mono-gram (English)**	**Importance**
I	I	0.027101	ベクトル	Vector	0.012692
III	III	0.026868	数	Number	0.011921
II	II	0.025151	関数	Formula	0.011126
A	A	0.018989	確率	Probability	0.009463
B	B	0.016693	複素	Complex	0.008825
数	Number	0.012049	点	Point	0.008570
関数	Function	0.011044	式	Formula	0.008534
解説	Solution	0.010463	よっ	Therefore	0.008256
ベクトル	Vector	0.009900	極限	Limit	0.008139
点	Point	0.009046	定積	Constant volume	0.007628
確率	Probability	0.008046	列	Column (or Sequence)	0.006980
複素	Complex	0.007726	値	Value	0.006849
式	Formula	0.007594	次	Next	0.006669
列	Column (or Sequence)	0.006998	求め	Find [the value]	0.006547
積分	Integral	0.006740	する	Do	0.006483
三角	Triangle	0.006716	積分	Integral	0.006374
定積	Constant volume	0.006563	三角	Triangle	0.006342
極限	Limit	0.006439	が	Be	0.006325

addition, a previous study documented good results when using the Bag-of-Words method and Random Forest (Montoliu et al., 2015). This is also possibly a reason why this model yielded the best performance.

Since Random Forest uses decision trees, it is easy to create accurate decision techniques for binary vectors. Therefore, we believe that classification using Random Forest was able to accurately classify binary vectors with numerous dimensions. In addition,

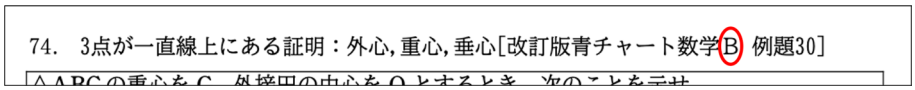

Fig. 13 The top of a sentence in a PDF. There is a character representing the part of organization in the first line (in this example, 'B' surrounded by a red circle)

the fact that characters such as 'I', 'II', 'III', 'A', and 'B' existed as typical classification indices in Japanese mathematics and had a significant influence on classification, suggests that Random Forest with mono-grams produced the best prediction accuracy. This also indicates that the organization characters can be useful when classifying the exercises: if the PDF sentence contains such characters, as shown in Fig. 13, it can be easier to automatically classify it.

Selection of evaluation method in the educational context

Three indices, A_L, F_L, and F_{wL}, were used in the experiment as evaluation indices. A_L is desired to be evaluated with a more reliable index, since in the present data set, there are much more data that are true-negative than true-positive data and tend to rate the model that is false for all data highly (Manning et al., 2008). In this case, the indicator F-measure is often used for two-class classification, but there are two ways to obtain it for multi-class classification. In the case of multiclass classification, there are two ways to obtain the F-measure: F_L and F_{wL}.

F_L returns the average of the F_l obtained for each class l, which is equal to the average of the F_l obtained for all classes, even if the number of data in each class is not uniform. Therefore, it is possible to treat all units equally even if the number of data in each unit is not uniform. In other words, it is an effective indicator for labeling biased data sets. For example, F_L is useful when a teacher in a school setting selects three 1st level units to create a test (ignoring the rest) and automatically assigns 2nd level units to exercises within those units. F_{wL} is the F-measure calculated from the sum of Precision P_l and Recall R_l for each class l. This is a weighted index that takes into account the number of each data set. Therefore, it can be said that the index accurately reflects the distribution of this data set. The index F_{wL} is useful for labeling a uniform data set, i.e., the mathematical material studied in three years of Japanese high school at a time.

From the experimental results, we can say that the combination of mono-gram and Random Forest, which has the largest F_L, is effective when limiting the unit, and the combination of bigram and Random Forest is effective for a uniform data set of three years of high school. However, the accuracy rate is not much different when using the feature $n = 1$ or $n = 2$ in Random Forest prediction.

Practical educational implications in this research

Automatic labeling can help reduce teacher workload (Tian et al., 2022) and develop mathematics workmanship (Fishback & Schlicker, 1996) by introducing systems that require labels (Vie et al., 2019; Wang et al., 2022). For further development of mathematics in Japan, a programming environment on supporting units on mathematics using data (Kayama et al., 2022) and clarification of unit structure for knowledge association in learning (Taniguchi & Itoh, 2023) have been proposed.

The labeling method in this study allows labels to be assigned to unlabeled mathematical instructional materials by learning the text of the labeled materials, even though they are not formatted suitably for extracting text, such as handmade materials by mathematics teachers. These can facilitate unit learning based on the national standard curriculum guidelines. For example, when students study on their own, the system can suggest different exercises than the ones they have solved, with the explanation that they are part of the same unit, which can promote student understanding. Therefore, it can be said that a system using units is more easily utilized in the school contexts. In other words, the contribution of this study is that the automatic assignment of unit information to systems in the educational field will expand the range of support without burdening teachers.

In addition, although we have chosen to use mathematical exercises as the subject matter, we believe that such a method could be applied to other subjects as well, given the uniform treatment of equations, terminology, and other information as textual information. To do so, we need clearly shared criteria and examples of exercises to which they are pre-assigned (i.e., we can use the method proposed in Sect. 3 if the data set is in a usable format).

Limitations and future research

In this study, we proposed an algorithm to solve the problem of classifying incomplete texts of mathematical exercises into different leveled units. However, if more detailed text were available, a context-aware classification algorithm is expected to produce better accuracy.

In this experiment, we limited ourselves to one topic of the same level to be assigned to each mathematical exercise, but there also are mathematical exercises that span multiple topics. In order to properly assign topics to such mathematical exercises, a system that can assign multiple units using the algorithm verified in this study is needed. Multi-label classification is also widely used in machine learning (Sorower, 2010; Tsoumakas & Katakis, 2007). Once such a system is completed, it would be possible to recommend similar exercises using mathematical topics and analyze student learning based on topics.

This experiment showed that even if it is not possible to read mathematical expressions, numbers, or symbols, it is possible to classify with high accuracy using only the textual information obtained. Once such a system is developed, it would be possible to recommend similar exercises based on mathematical topics and analyze student learning based on topics. Since the system would be able to assign common labels to different teaching materials, it would be possible to develop a textbook recommendation system that assigns textbook subsections to exercises so that students can review them in the textbook when they make a mistake on an exercise. The information collected using these systems would then create a learning support environment that takes into account the degree of difficulty and understanding of the 1st level unit and 2nd level unit itself.

In addition, since the experiment was conducted independently of student learning, there are no results on the contribution to learner and teacher activities. It will be necessary to verify the educational effects in future experiments by using the automatic labeling of the unit to recommend teaching materials or to analyze the behavior of the learners, especially with predicting entire difficulty of the exercises and learners'

complete rate of them, or considering students reading comprehension. In addition to this, as one study developed a recommendation system which uses student's action as a parameter of the system (Takami et al., 2022), there is also room for combining following two approaches: topic-based model driven approach (i.e., the labeling the exercises) and student's behavior data driven approach (i.e., using the student's achievement of the exercise into the system).

Conclusion

This paper proposes an algorithm that uses several techniques to correctly assign topics to the incomplete mathematical text obtained from PDF text. The extracted text showed that all information on numbers, mathematical expressions, and symbols was omitted when converted from PDF to text. Furthermore, we compared the prediction accuracy of the two methods at the stage of predicting topics from the obtained vectors. Two methods were used to compare their prediction accuracy: one using cosine similarity and the other using machine learning. We attempted to predict with all features and models and found that the best prediction accuracy was achieved by using mono-grams as features and applying Random Forest (92.5% and 68.5% for 1st level unit and 2nd level unit, respectively). We conclude that the reason for the higher accuracy was the ability to find context-independent similarities even in incomplete sentences by using n-grams to find matches in which the remaining words are used, and the existence of organization parts ('I', 'II', 'III', 'A', 'B') representing common national classifications for Japanese mathematical exercises. Given that PDFs are not necessarily assigned such national symbols, we conducted a similar experiment omitting them as stop words and found that the accuracy dropped a little, but important mathematical knowledge elements appeared in the key features, which are important for the classification of mathematical exercises.

The contribution in the research is the discovery that mono-grams, a simpler approach similar to traditional methods like n-grams or bag of words, outperformed state-of-the-art methods in classifying incomplete texts, particularly in the context of Japanese mathematics exercises. These findings challenge previous research results and suggests that the choice of text analysis techniques may depend on the specific task or target domain.

Abbreviations

ACARA	Australian curriculum, assessment and reporting authority
BERT	Bidirectional encoder representations from transformers
CCSS	Common core state standard
CNN	Convolutional neural network
ELMo	Embeddings from language models
GPT	Generative pretrained transformer
HTC	Hierarchical text classification
HTML	Hyper text markup language
ICT	Information and communication technology
K-12	Kindergarten through 12th grade
LR	Logistic regression
LSTM	Long short term memory
MEXT	Ministry of Education, Culture, Sports, and Technology
MLP	Multi layered perceptron
MOE	Ministry of Education
MSC	Mathematics subject classification
OCR	Optical character recognition
PDF	Portable document format
PISA	Programme for international student assessment
RF	Random forest

RNN	Recurrent neural network
SES	Socio-economic status
US	United States
VSM	Vector space model
WE-KE	Word embedding and knowledge extracting
XGB	EXtreme gradient boosting
zbMATH	Zentralblatt MATH

Author contributions
RN, BF, and HO contributed to the research conceptualization and methodology. TY wrote the manuscript. RN, YD, KT, BF, and HO provided comments to improve the manuscript. All authors read and approved the final manuscript.

Funding
This work was partly supported by JSPS Grant-in-Aid for Scientific Research (B) JP23H01001, JP22H03902, JP20H01722, JSPS Grant-in-Aid for Scientific Research (Exploratory) JP21K19824, and NEDO JPNP20006.

Availability of data and materials
The data of this study is not open to the public due to participant privacy.

Declarations

Competing interests
The author declares no competing interests.

Received: 4 July 2023 Accepted: 9 October 2023
Published online: 18 October 2023

References

Abekawa, T., & Aizawa, A. (2016). SideNoter: Scholarly paper browsing system based on PDF restructuring and text annotation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations,* 136–140.

Australian Curriculum, Assessment and Reporting Authority (ACARA). *F-10 curriculum mathematics structure*. Retrieved 01 September, 2023 from https://www.australiancurriculum.edu.au/f-10-curriculum/mathematics/structure/.

Bhartiya, D., Contractor, D., Biswas, S., Senjupta, B., & Mohania, M. (2016). Document segmentation for labeling with academic learning objectives. In *Paper presented at the International Conference on Educational Data Mining,* 282–287.

Breiman, L. (2001). Random forests. *Machine Learning, 45*, 5–32.

Cavnar, W. B., & Trenkle, J. M. (1994). N-gram-based text categorization. In *Proceedings of SDAIR-94, 3rd annual symposium on document analysis and information retrieval*, 161175.

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794.

Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., & Chen, K. (2015). Xgboost: Extreme gradient boosting. *R Package Version, 1*(4), 1–4.

Chow, J. C., & Ekholm, E. (2019). Language domains differentially predict mathematics performance in young children. *Early Childhood Research Quarterly, 46*, 179–186.

Church, K. W. (2017). Word2Vec. *Natural Language Engineering, 23*(1), 155–162.

Contractor, D., Popat, K., Ikbal, S., Negi, S., Sengupta, B., & Mohania, M. K. (2015). Labeling educational content with academic learning standards. In *Proceedings of the 2015 SIAM International Conference on Data Mining*, pp. 136–144.

Cox, D. R. (1958). The regression analysis of binary sequences. *Journal of the Royal Statistical Society: Series B (methodological), 20*(2), 215–232.

Date, I., & Isozaki, H. (2015). Detection of mathematical formula regions in images of scientific papers by using deep learning and OCR. *IEICE Technical Report, 2015*(4), 1–6. in Japanese.

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 1*, 4171–4186.

Dharma, E. M., Gaol, F. L., Warnars, H. L. H. S., & Soewito, B. (2022). The accuracy comparison among word2vec, glove, and fasttext towards convolution neural network (CNN) text classification. *Journal of Theoretical and Applied Information Technology, 100*(2), 349–359.

Ditchburn, G. (2012). A national Australian curriculum: In whose interests? *Asia Pacific Journal of Education, 32*(3), 259–269.

Dunne, E., & Hulek, K. (2020). Mathematics subject classification 2020. *EMS Newsletter, 115*, 5–6.

Fateman, R. J., Tokuyasu, T., Berman, B. P., & Mitchell, N. (1996). Optical character recognition and parsing of typeset mathematics. *Journal of Visual Communication and Image Representation, 7*(1), 2–15.

Fishback, P., & Schlicker, S. (1996). The impact of technology on mathematics education. *Grand Valley Review, 14*(1), 27.

Flanagan, B., Majumdar, R., Akçapınar, G., Wang, J., & Ogata, H. (2019). Knowledge map creation for modeling learning behaviors in digital learning environments. In *Companion Proceedings of the 9th International Conference on Learning Analytics and Knowledge,* 428–436.

Flanagan, B., & Ogata, H. (2018). Learning analytics platform in higher education in Japan. *Knowledge Management & E-Learning: An International Journal, 10*(4), 469–484.

Graovac, J. (2014). Text categorization using n-gram based language independent technique. *Intelligent Data Analysis, 18*(4), 677–695.
Graovac, J., Kovačević, J., & Pavlović-Lažetić, G. (2015). Language independent n-gram-based text categorization with weighting factors: A case study. *Journal of Information and Data Management, 6*(1), 4–17.
Graovac, J., Kovačević, J., & Pavlović-Lažetić, G. (2017). Hierarchical vs. flat n-gram-based text categorization: Can we do better? *Computer Science and Information Systems, 14*(1), 103–121.
Guo, Y., Silver, E. A., & Yang, Z. (2018). The latest characteristics of mathematics education reform of compulsory education stage in China. *American Journal of Educational Research, 6*(9), 1312–1317.
Hussein, H. B. (2023). Global trends in mathematics education research. *International Journal of Research in Educational Sciences., 6*(2), 309–319.
Ikeda, T. (2021). nagisa (0.2.7). https://github.com/taishi-i/nagisa.
Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2017). Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, 2*, 427–431.
Kayama, M., Nagai, T., & Asuke, T. (2022). A proposal for a visual programming environment for "use of data" related units in primary and secondary education. *Journal of Japanese Society for Information and Systems in Education, 39*(2), 224–234. in Japanese.
Khan, A., Baharudin, B., Lee, L. H., & Khan, K. (2010). A review of machine learning algorithms for text-documents classification. *Journal of Advances in Information Technology, 1*(1), 4–20.
Khosravi, H., & Cooper, K. (2018). Topic dependency models: Graph-based visual analytics for communicating assessment data. *Journal of Learning Analytics, 5*(3), 136–153.
Kobayashi, T. (2021). T-vMF similarity for regularizing intra-class feature distribution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,* 6616–6625.
Kobayashi, Y., Tanaka, S., & Tomiura, Y. (2012). Pattern recognition of english scientific papers using n-grams. *Information Fundamentals and Access Technologies, 12*(1), 1–6.
Kühnemund, A. (2016). The role of applications within the reviewing service zbMATH. *PAMM, 16*(1), 961–962.
Li, B., Liu, T., Du, X., Zhang, D., & Zhao, Z. (2016). Learning document embeddings by predicting n-grams for sentiment classification of long movie reviews. In *The Eleventh International Conference on Learning Representations*.
Liu, G., & Guo, J. (2019). Bidirectional LSTM with attention mechanism and convolutional layer for text classification. *Neurocomputing, 337*, 325–338.
Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge University Press.
Mansur, M. (2006). *Analysis of n-gram based text categorization for Bangla in a newspaper corpus* (Doctoral dissertation, BRAC University).
Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Proceedings of Advances in Neural Information Processing Systems 26*.
Ministry of Arts and Sciences (MEXT). (2009). 高等学校学習指導要領(平成*21*年*3*月告示*). [High school curriculum guidelines (announced in March 2009)]*. https://erid.nier.go.jp/files/COFS/h20h/index.htm. in Japanese.
Ministry of Arts and Sciences (MEXT). (2018). 数学編・理数編 高等学校学習指導要領(平成*30*年告示*). [In mathematics and science, high school curriculum guidelines (announced in 2018)]*. https://www.mext.go.jp/content/20230217-mxt_kyoiku02-100002620_05.pdf. in Japanese.
Ministry of Arts and Sciences (MEXT). (2021). 高等学校用教科書目録(令和*4*年度使用*) [For higher education textbook catalog (for fiscal year 2021)]*. https://www.mext.go.jp/content/20210604-mxt_kyokasyo02-000014470_4.pdf. in Japanese.
Ministry of Education of the People's Republic of China (MOE). (2012). *Mathematics curriculum standards for compulsory education* (2011th ed.). Beijing Normal University Press.
Montoliu, R., Martín-Félez, R., Torres-Sospedra, J., & Martínez-Usó, A. (2015). Team activity recognition in association football using a bag-of-words-based method. *Human Movement Science, 41*, 165–178.
Ohnishi, T. (2011). Task-based learning in high school mathematics. *Japan Society for Science Education Research Report, 26*(8), 45–48. in Japanese.
Palmer, J. A. (2021). pdftotext (2.2.2). https://github.com/jalan/pdftotext.
Peters, M., Neumann, M., Zettlemoyer, L., & Yih, W. (2018). Dissecting contextual word embeddings: Architecture and representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 1499–1509.
Porter, A., McMaken, J., Hwang, J., & Yang, R. (2011). Common core standards: The new US intended curriculum. *Educational Researcher, 40*(3), 103–116.
Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI Blog, 1*(8), 9.
Raff, E., Richard, Z., Cox, R., Sylvester, J., Yacci, P., Ward, R., Tracy, A., Mclean, M., & Nicholas, C. (2018). An investigation of byte n-gram features for malware classification. *Journal of Computer Virology and Hacking Techniques, 14*, 1–20.
Ramakrishnan, C., Patnia, A., Hovy, E., & Burns, G. A. (2012). Layout-aware text extraction from full-text PDF of scientific articles. *Source Code for Biology and Medicine, 7*(1), 1–10.
Ritter, B. J. (2009). *Update on the common core state standards initiative*. National Governors Association.
Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review, 65*(6), 386.
Schubotz, M., Scharpf, P., Teschke, O., Kühnemund, A., Breitinger, C., & Gipp, B. (2020). Automsc: Automatic assignment of mathematics subject classification labels. In *International Conference on Intelligent Computer Mathematics,* 237–250.
Shen, J. T., Yamashita, M., Prihar, E., Heffernan, N., Wu, X., McGrew, S., & Lee, D. (2021). Classifying math knowledge components via task-adaptive pre-trained BERT. In *International Conference on Artificial Intelligence in Education,* 408–419.
Shintani, R. (2014). The development process and contents of the common core state standards: Based on a comparative study with the Japanese Course of Study for lower secondary school. *Journal of Japan Association of American Educational Studies, 25*, 15–27. in Japanese.
Silla, C. N., Jr., & Freitas, A. A. (2011). A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery, 22*(1–2), 31–72.

Smith, R. (2007). An overview of the tesseract OCR engine. In *Ninth International Conference on Document Analysis and Recognition, 2*, 629–633
Sorower, M. S. (2010). A literature survey on algorithms for multi-label learning. *Oregon State University, Corvallis, 18*(1), 25.
Sosnovsky, S., & Brusilovsky, P. (2015). Evaluation of topic-based adaptation and student modeling in quizguide. *User Modeling and User-Adapted Interaction, 25*, 371–424.
Spelke, E. S., & Tsivkin, S. (2001). Language and number: A bilingual training study. *Cognition, 78*(1), 45–88.
Suen, C. Y. (1979). N-gram statistics for natural language understanding and text processing. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 2*, 164–172.
Takami, K., Dai, Y., Flanagan, B., & Hiroaki Ogata. (2022). Educational explainable recommender usage and its effectiveness in high school summer vacation assignment. In *12th International Learning Analytics and Knowledge Conference,* 458–464.
Taniguchi, Y., & Itoh, T. (2023). Unit association method with symbolization in high school mathematics textbook. *Journal of Information Processing, 64*(1), 256–269. in Japanese.
Tian, Z., Flanagan, B., Dai, Y., & Ogata, H. (2022). Automated matching of exercises with knowledge components. In *30th International Conference on Computers in Education Conference Proceedings,* 24–32.
Tsoumakas, G., & Katakis, I. (2007). Multi-label classification: An overview. *International Journal of Data Warehousing and Mining, 3*(3), 1–13.
Vie, J. J., & Kashima, H. (2019). Knowledge tracing machines: Factorization machines for knowledge tracing. *Proceedings of the AAAI Conference on Artificial Intelligence, 33*(1), 750–757.
Vovides, Y., Sanchez-Alonso, S., Mitropoulou, V., & Nickmans, G. (2007). The use of e-learning course management systems to support learning strategies and to improve self-regulated learning. *Educational Research Review, 2*(1), 64–74.
Wang, F., King, R. B., & Leung, S. O. (2023). Why do east Asian students do so well in mathematics? A machine learning study. *International Journal of Science and Mathematics Education, 21*(3), 691–711.
Wang, J., Minematsu, T., Okubo, F., & Shimada, A. (2022). Topic-wise representation of learning activities for new learning pattern analysis. In *30th International Conference on Computers in Education Conference Proceedings, 1*, 268–278.
zbMATH OPEN, The first resource for mathematics. *Mathematics subject classification—MSC2020.* Retrieved 03 September, 2023 from https://zbmath.org/classification/.
Zheng, E., Moh, M., & Moh, T. S. (2017). Music genre classification: A n-gram based musicological approach. In *2017 IEEE 7th International Advance Computing Conference,* 671–677.

Publisher's Note