Improved Automated Labeling of Mathematical Exercises in Japanese

Taisei YAMAUCHI[a*], Ryosuke NAKAMOTO[a], Yiling DAI[b], Kyosuke TAKAMI[b,c], Brendan Flanagan[d], & Hiroaki OGATA[b,c]

[a]*Graduate School of Informatics, Kyoto University, Japan*

[b]*Academic Center for Computing and Media Studies, Kyoto University, Japan*

[c]*Education Data Science Center, National Institute for Educational Policy Research (NIER), Japan*

[d]*Center for Innovative Research and Education in Data Science, Institute for Liberal Arts and Sciences, Kyoto University, Japan*

*yamauchi.taisei.28w@st.kyoto-u.ac.jp

Abstract: This study aims at improving the prediction quality of the automatic labeling of learning materials. Labeling learning materials has two existing issues: establishing completely automated labeling and reducing manual labor for assigning labels to materials. Labels of the materials are utilized for analyzing students' learning patterns, tracing knowledge, and recommending exercises to students. Since it is too burdensome to manually assign several labels to many learning materials, an automatic, algorithm-based labeling system is desirable. However, classification using word embedding has often yielded lower accuracy for mathematics learning materials with short texts. In this research, we have conceived and implemented an improved approach to predict a label by calculating the similarity of n-gram of sentences using Jaccard coefficients, weighting them to create a vector representation, and using it to predict the label of the exercise. We compared the accuracy and F score of the prediction results of the weighted n-gram similarity model with those of the state-of-the-art word embedding model. We found that the n-gram approach was superior in both accuracy and F score. Furthermore, we plotted the vectors obtained from each model in two-dimensional coordinates and observed that the n-gram model produced more flexible predictions, regardless of the vector's position. These results suggest the classification effectiveness with weighted similarity of n-gram for materials with a small amount of text.

Keywords: Automatic labeling, word n-gram, Jaccard coefficient, t-SNE, word embedding, topic-based learning.

1. Introduction

Labeling learning materials is a widely used method in educational domains, such as knowledge tracing to adapt how to teach to each student (Vie & Kashima, 2019), automatic exercise recommendations to support students' learning strategy (Takami et al., 2022) and analysis based on the material topic (Wang et al., 2022). With the shift to ICT education, to predict how well the students tackle exercises (Vie & Kashima, 2019), to recommend the most appropriate exercise in many learning materials (Takami et al., 2022) or to utilize them for learning pattern analysis (Wang et al., 2022), a topic-based classification of instructional materials is critical to utilize them conveniently. However, assigning the classification to problems manually is a hard task that requires the cooperation of experts. In other words, it is desirable to label exercises automatically to reduce the cost and time of domain experts who usually perform the task. Previous classification studies have taken one reliable method: word embedding. A previous study has shown that word embedding has produced novel results for classifying news stories (Dharma et al., 2022). However, another previous research into mathematical exercise classification results in lower accuracy for exercises, each consisting of a small number of words (Tian et al., 2022).

In this study, we propose an automatic classification algorithm that can handle even short sentences in mathematics exercises. We focused on the optimal agreement of a set of mathematical problems and calculated the similarity between the weighted word n-gram variance representation of any labeled exercise and that of each exercise query. For further understanding of the results, we use t-SNE (Van der Maaten & Hinton, 2008) and compared the proposed bi-gram vector space model with that of novel word embedding. We set the following research question and tackle it:

RQ: Can the use of a weighted n-gram classifier improve the labeling of Japanese mathematics quizzes compared to previous research using state-of-the-art word embedding methods?

2. Literature Review

There is a trend toward analyzing learning behavior in a new way using the labels assigned to teaching materials. With regard to the use of features in learning effectiveness analysis, a study reported that the proposed system automatically assigned labels with learning materials and analyzed the assigned labels to discover students' learning patterns (Wang et al., 2022). Giving metadata 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 & Kashima, 2019). It is also useful to categorize many exercises for recommending a specific exercise to enhance students' understanding. One study developed a recommendation system with explanations based on an explanation generator using parameters from a Bayesian knowledge tracing model (Takami et al., 2021). The recommendation quiz with explanations was used more frequently and with more continuous users compared to a system without explanations (Takami et al., 2022). There has also been researching into extracting labels from learning materials to form knowledge structure representations that can be shown to learners to increase their awareness of the study process (Flanagan et al., 2019). These research examples show the importance of providing labels to materials because it is easier to obtain or utilize detailed information about the characteristics of the material if they are labeled in advance.

Studies on automated labeling or classifying methods also focus on reducing the burden of manual labeling. A study has attempted to classify using 385 different labels to classify 12 years of mathematics materials from kindergarten through high school (Shen et al., 2021b), whose classification categories are according to the common core standard developed by the United States (Ritter, 2009). The label of the study is for K-12 Education in the United States, which is not somewhat appropriate for the Japanese one. One study proposed the MathBERT model, a model created by pre-training the base BERT model (Devlin et al., 2018) on a large corpus of mathematics content ranging from kindergarten students (pre-k) to high school and university graduate-level mathematical content (Shen et al., 2021a). Labeling and classifying instructional materials automatically is receiving much attention, and several researchers have investigated methods to solve this important task. One paper proposed an automatic classification method in a mathematical subject classification scheme for organizing mathematical literature, achieving a classification accuracy of 81%, which is very close to the classification accuracy of two extensive peer-review services. It also enabled an 86% labor reduction compared to manual classification (Schubotz et al., 2020). This study shows the importance of labeling materials automatically to reduce the human labor of manual labeling.

Previous research has found that word embedding can be effectively applied to label classification tasks. Dharma, et al. (Dharma et al., 2022) found that word embedding using the Fasttext method was able to classify a dataset containing 19,977 news stories and 20 news topics with 97.2% accuracy, which is more accurate than any other word embedding method.

However, when it comes to classifying exercises with short sentences, sentence vectorization using word embedding has been found to be a less effective method. Tian et al. (Tian et al., 2022) applied word embedding as a method to classify short Japanese exercise texts and found it achieved 72.87% accuracy. Accuracy was then further improved by combining the method in an ensemble with a keywords extraction method. This suggests that the word embedding method might not be as effective for short exercise texts.

Therefore, in this study, we propose an automatic classification algorithm that can handle even short Japanese sentences in mathematics exercises by focusing on the optimal agreement of a set of mathematical problems by calculating the similarity between the weighted word n-gram variance representation of any labeled exercise text and the query exercise texts.

3. Method

While word embedding has been effectively applied to many tasks, labeling mathematics exercises in Japanese that contain few words is challenging and has not performed as anticipated. In this paper, in addition to classification by word embedding, we examine the classification by Jaccard coefficient word n-gram to see if it can provide improved performance in the task. For this purpose, we assigned an appropriate unit label to the exercises, trained them using an algorithm, and made predictions.

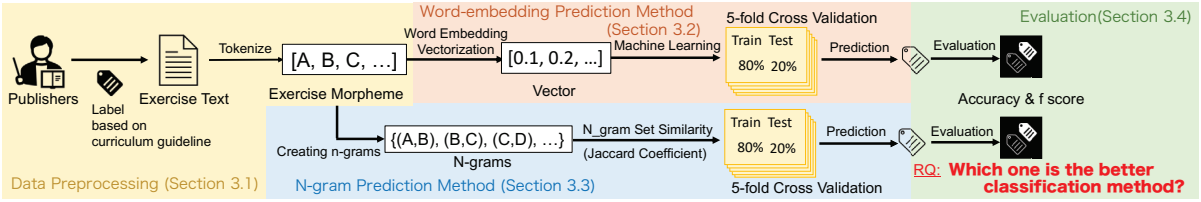

Figure 1. Overview of the method.

The details of the two approaches will be explained in the following sections as indicated in Figure 1. Note that we explain the details of the two approaches in the following sections.

3.1 Data Preprocessing

The exercise texts used in the experiment are from six different digital mathematical exercise books. These exercise books are for high school students in accordance with the textbooks designated by the Japanese Ministry of Education, Culture, Sports, Science and Technology (MEXT, 2021), and are made by the same company that produced the textbooks, which are all well-used in Japanese. We prepared text files by reading text data from pdf using the Python library Pdf2text (Palmer, 2021). Due to possible problems from file structure and text embedding methods, Pdf files are more difficult to obtain in their complete text form than HTML-formatted files, as previous studies confirm (Ramakrishnan et al., 2012; Smith, 2007).

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. This process is required to extract individual words because there are no explicit word separators such as blank spaces in Japanese. When Nagisa morphologically analyzes number digits, it recognizes a set of digits that each recognized as a noun. For treating different numbers as the same, we processed a set of digits into a single number and treated all numbers as the same thing.

Each exercise has the subject labels whose publisher allocated on, so we treated them as the correct labels. There was a total of 2,775 exercises, consisting of 24 subject labels, and each subject label was assigned between 25 and 200 exercises. Figure 2 shows an example of an exercise pdf data and the assigned subject label. The figure also shows a pair of exercise

text data with very similar sentences, and other pairs of problem data with similar sentences were found in the data set.

There are 24 subjects, with each having an average of 115.6 exercises. Each exercise contains M = 98.5, and SD = 61.4 morphemes, with a minimum of 17 and a maximum of 469 occurring in the dataset that was collected. This shows the limited amount of information contained in the dataset which the classifier can use to automatically assign labels. Compared to English, mathematics quizzes the number of morphemes is smaller.

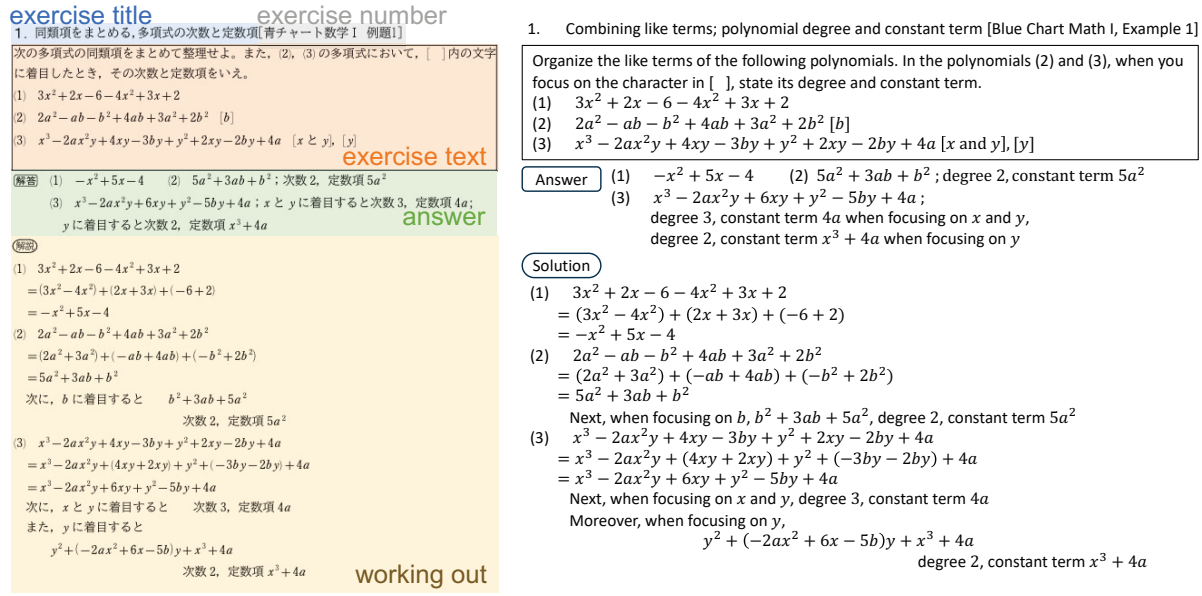

Figure 2. Example of exercises in the dataset (Right: English translation). The subject label "number and formula" is assigned to an exercise in the figure.

3.2 Existing Word-embedding Prediction Method

For vectorization with word embedding, we used a model called fastText (Joulin et al., 2016). Among the fastText models for 157 languages, we used the Japanese model in this experiment. This model combines three methods to represent a single sentence data in 300 dimensions: character 5-gram, weighting by position, and Word2Vec (Church, 2017). By considering each word as a set of its constituent sub-words, it offers notably improved performance when dealing with languages that have complex morphological structures including Japanese (Khan et al., 2022). To evaluate classification performance using word embedding, we used 5 baseline models which have different characters. Five models we used are as follows:

- XGBoost (Chen & Guestrin, 2016): a model that combines boosting and decision trees and has shown good results in various natural language processing tasks, so it's proper to use in the context of this paper.
- Random Forest (Breiman, 2001): a model that uses many decision trees trained by randomly sampled training data. This works well even with a large number of explanatory variables, so it can deal with a 300-dimensional vector.
- Support Vector Machine (Vapnik and Lerner, 1963): a model for constructing a two-or-more-class pattern discriminator using linear input elements. This is one of the most popular methods to classify something, so it is appropriate to use as a baseline.
- Logistic Regression (Cox, 1958): a statistical regression model with variables that follow a Bernoulli distribution. As well as SVM, it is one of the most popular methods to classify something, so it is appropriate to be treated as a baseline.
- Multilayer Perceptron (Gardner & Dorling, 1998): a model that uses feedforward artificial neural networks. A neural network is also treated as the baseline for the machine learning tasks, so we use it in this experiment.

3.3 Proposed N-gram Prediction Method

We created the word n-gram from the tokenized exercise sentences. The prepared data set is divided into label data and query (query means an unlabeled exercise), and the similarity between the set of word n-grams in the labeled data and the set of word n-grams in the query data is calculated. We described how to choose one subject for each not-yet-labeled exercise in Figure 3. In the figure, $n_{t_{A_k}}$s in green boxes represent the n-gram set of the labeled exercise text, and n_{t_q}s in blue boxes do that of a query. We evaluated the similarity $J_{l x_{k,q}}$ between $n_{t_{A_k}}$ and n_{t_q} by calculating the Jaccard coefficients of those. Then, we weighted $J_{l x_{k,q}}$s and found a value $J_{l,q}$ corresponding to label l. If the value $J_{l,q}$ is the highest of all l of $J_{l,q}$, then the label l was predicted.

For vectorization using the word n-gram, we used a function in our algorithm. As previous studies improve accuracy by weighting for realistic non-homogeneous data sets (Graovac et al., 2015; Kobayashi, 2021), we modified the weights of the calculated n-gram similarity to create a more accurate classifier. We created two functions based on two assumptions as follows:

- Assumption 1: Two exercises that have the same label are similar to each other. Therefore, we created $f_{mean}\ (1)$ to find the most proper label considering all labeled exercises' similarity.
- Assumption 2: m specific exercises with the same label have high similarity with each other. Therefore, we created $f_{rank_m}\ (2)$ to find the most proper label considering m labeled exercises' similarity.

We substitute $J_{l_{k,q}}$ for all labeled data l_k with label l all into the determined function f to obtain the weight coefficient $J_{l,q}$ of queries q that are assigned that label. Hereafter $\mathrm{LARGE}(J_{l,q}, k)$ represents the kth highest value of $J_{l_1,q}, J_{l_2,q}, \ldots, J_{l_n,q}$. In this experiment, the following functions were defined to determine a more suitable weighting for classification:

$$f_{mean} = \frac{\sum_{k=1}^{n} J_{l_k,q}}{n} \tag{1}$$

$$f_{rank_m} = \frac{\sum_{k=1}^{m}\left((m-k+1)\times \mathrm{LARGE}(J_{l,q},k)\right)}{\sum_{k=1}^{m} k} \tag{2}$$

We then predict a label $l_{pred,q}$ for each query exercise q by choosing the highest coefficient of $J_{l,q}$:

$$l_{pred,q} = \underset{l}{\mathrm{argmax}} J_{l,q} \tag{3}$$

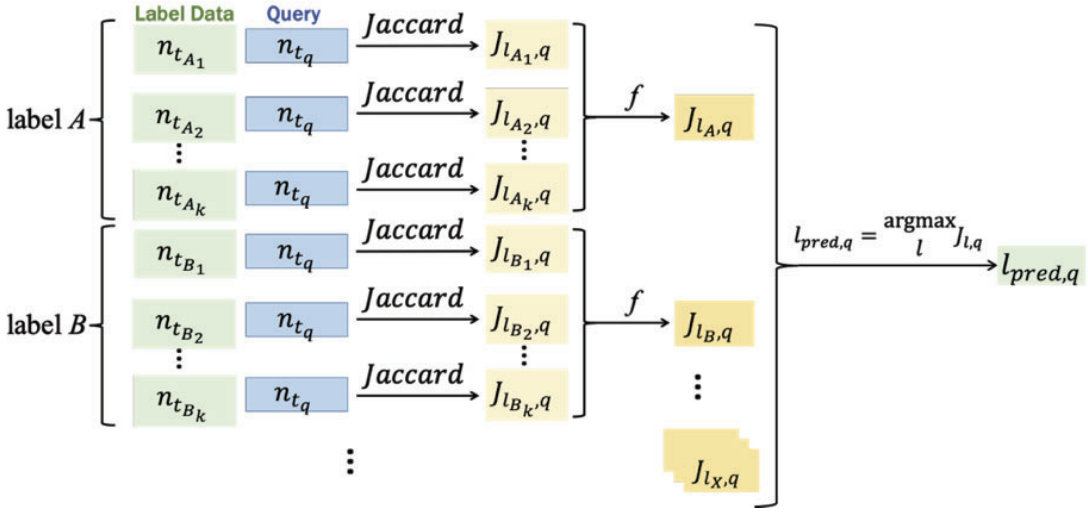

Figure 3. The way to find the weighted coefficient and predict a label.

3.4 Evaluation

To prevent bias in the dataset due to random sampling, we performed classification training with 80% of the entire dataset as training data and the rest as test data using 5-fold, a form of cross-validation. This allows us to train five times on one dataset.

To evaluate the effectiveness of our proposed method, we compare it to existing word embedding methods using the following two metrics: accuracy and F score. Accuracy and F scores on binary predictions are typically calculated by accurate prediction divided by all predictions and the harmonized average of precision and recall, respectively, but since the model in this study proposed here performs multi-class classification, it is necessary to use a calculation method customized for multi-class classification. For evaluation of the result in this study, the accuracy A_L and the precision P_L, the recall R_L, and the F score F_L for all labels were calculated using the following equation (Sorower, 2010).

$$A_L = \frac{\sum_{l \in L} A_l}{n(L)}, P_L = \frac{\sum_{l \in L} P_l}{n(L)}, R_L = \frac{\sum_{l \in L} R_l}{n(L)} \quad (4)$$

$$F_L = \frac{2P_L R_L}{P_L + R_L} \quad (5)$$

For further examine the experimental results, we also mapped the assigned labels to a two-dimension space for visualization by using t-SNE. The technique t-SNE is one novel technique that can project high-dimensional data into two dimensions and map them (Van der Maaten & Hinton, 2008). One study compared how well the clustering of the different methods with observing the visualization of the result with t-SNE, and found that one model that uses sentence embedding with knowledge graph embedding outperforms another model that only uses sentence embedding (Chen et al., 2022).

4. Result

4.1 Prediction Result

Figure 4 shows a plot of the accuracy for each weight function f for each value of n in the n-gram. We understand from the figure that the best accuracy was achieved for bigram ($n = 2$) for all functions f. Table 1 shows the accuracy and F score for each weight function for bigram. We found that the proposed algorithm predicts the best when we use the function f_{rank7}.

Table 1. *Relationship between weight function and accuracy in bigram.*

Function f	Accuracy	F score
f_{mean}	.7222	.7322
f_{rank1}	.8061	.8023
f_{rank2}	.8209	.8154
f_{rank3}	.8274	.8224
f_{rank4}	.8324	.8288
f_{rank5}	.8382	.8346
f_{rank6}	.8447	.8408
f_{rank7}	**.8483**	**.8441**
f_{rank8}	.8458	.8420
f_{rank9}	.8461	.8427
f_{rank10}	.8461	.8432

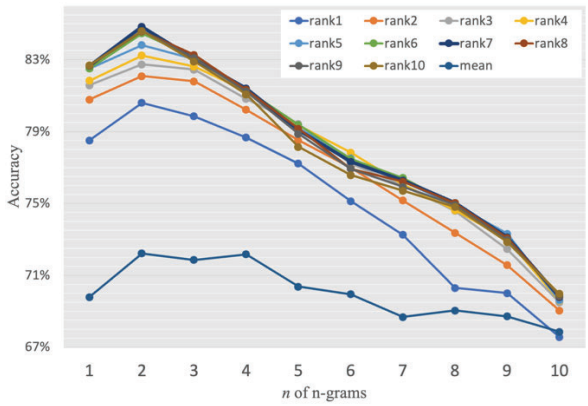

Figure 4. Relationship between n value in n-gram and accuracy with all models we used.

Table 2 shows the accuracy and F scores from each of these experiments using a word embedding and machine learning modules. We added the algorithm with the best classification accuracy in each n-gram experiment for comparison.

Table 2. *Accuracy and F score of the prediction using a word embedding with two essential results of the prediction using n-gram.*

Features	Models	Accuracy	F score
Word embedding	XGBoost	.6505	.6373
Word embedding	Logistic Regression	.6515	.6403
Word embedding	Perceptron	.6840	.6949
Word embedding	Random Forest	.7023	.6974
Word embedding	SVM	.7939	.7992
N-gram ($n = 2$)	Jaccard Coefficient + f_{mean}	.7222	.7322
N-gram ($n = 2$)	Jaccard Coefficient + f_{rank7}	**.8483**	**.8441**

4.2 Mapping Vector Representations with t-SNE

Figure. 5 shows the mapping of the results of classification using bi-gram (Jaccard + f_{rank7}) and that using word embedding (XGBoost). In the figure, exercises with different labels are represented by different colors and symbols. In (a)II, (b)II of the figure, exercises with wrong predictions are mapped in gray color.

Looking at (a)I and (b)I in Figure 5, we can see that both have a group of exercises with the same labels in close proximity to some extent. We can observe that the center of (a)I have a very large number of exercise data with different labels distributed tightly together, and from (a)II that they are almost always incorrectly predicted. However, (b)I do not show such portions of the data. This observation suggests that the word embedding method struggles to accurately label quizzes that are close to many different types of exercises when compared to using the proposed n-gram classification method.

Looking at (a)II and (b)II in Figure 5, we see that for both sets of data where the predictions are correct, exercises with the same label are distributed in close proximity to each other to some extent. In (a)II, there are many areas where items of the same type are clustered together and a few areas where items of different types are close to each other. On the other hand, it can be observed that in (b)II, there are many locations where items of different types are located close to each other. For example to compare with the label distribution in the transparent square of a(II) and b(II), for labels with labels a, c, d, k, o, and u, in (a)II, each label appears to distribute in a definite region, and each region appears to have no correct predictions for data with other labels. In contrast, in (b)II, we often find small regions where the labels are different but the predictions are correct. This accounts for the fact that prediction by word embedding is sensitive to the position corresponding to the vector representation of

the data, whereas n-gram similarity is able to make predictions regardless of the position corresponding to the vector representation of the data.

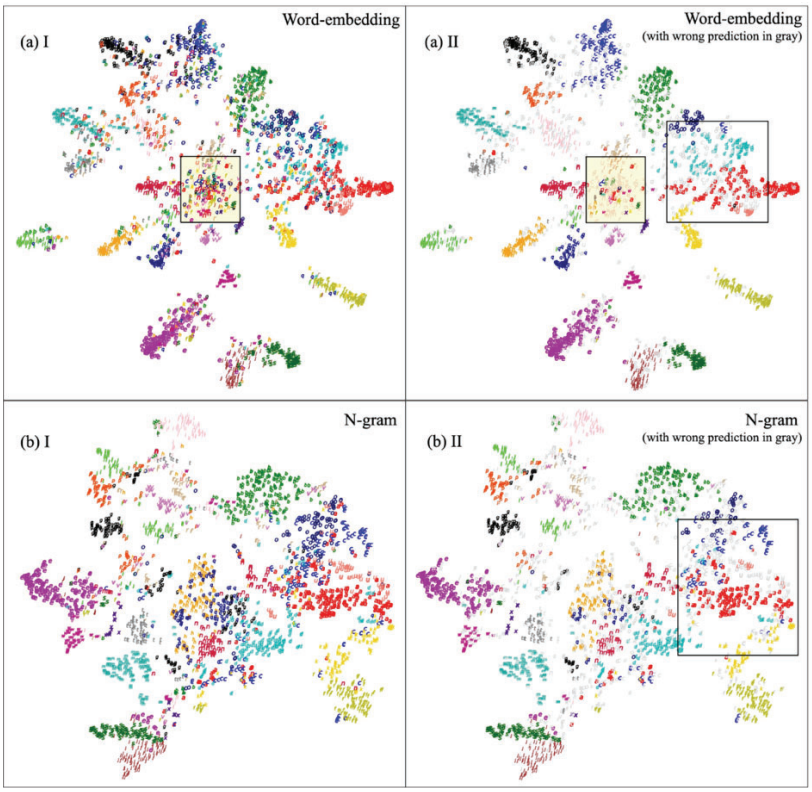

Figure 5. Maps of the vectors obtained from each experiment were transformed into two dimensions using t-SNE and taken on a plane.

5. Discussion and Limitations

The results in Table 1 show that prediction using a weighting function f_{rank7} is about 10% more accurate than classification using the function f_{mean}, and as the graph shows, we can say that other weight model also outperforms f_{mean} classification. This indicates that not all problems classified under the appropriate label are similar sentences and that learning all problems may result in noise. In the experiment, we classify the exercises into 24 classes with 115.6 questions per class, and the purpose of the weight function is to determine how many labeled data texts are like the query text. One piece of evidence supporting this is that there are often exercises in the data set that have very similar sentences and are given the same label. Thanks to the presence of such exercises, we were able to achieve a highly accurate multi-class classification even with a small data set.

We also found that the model with n-gram gave about 5% higher accuracy at the classification task, such as f_{rank7} against SVM. This indicates that a weighted n-gram similarity model is more effective for the classification of short math exercises.

In Figure 5 (a)I and (b)I, both models were observed to have regions assigned to units. Also, Figure 5 shows that the word embeddings model's map has regions assigned to the unit to some extent so that clumps of errors are noticeable in (a)II, but this is not the case for the prediction in (b)II. This indicates that classification using n-gram is more flexible in making predictions.

One limitation of this experiment is that although in this experiment we use two kinds of weighting function f, f_{mean} and f_{rank}, we can consider other weighting functions. Experimental results are enough to show the better performance of the n-gram representation model than the word embedding model, but we can expect another model yields a more accurate prediction.

6. Conclusion and Future Works

In this study, we propose an automatic classification algorithm that can handle even short sentences in mathematics exercises by focusing on the optimal agreement of a set of mathematical problems. We compared the proposed model which calculates the Jaccard coefficient, to evaluate the similarity of n-gram with the previously existing word embedding model. Experimental results show that the effectiveness in labeling mathematical exercises of the proposed algorithm using weighted n-gram exceeds that of word embedding, with an accuracy of 84.83% versus 79.39%, and an F score of 84.41% versus 79.92% respectively. We also compared the two methods using t-SNE and found that the proposed bi-gram vector space model made more flexible predictions regardless of the position of the vector transformed by t-SNE.

There are some limitations of the current study that should be mentioned, such as how the materials used for the experiment are all from only one publisher which could have lead to the high accuracy in the results. Additional experiments should be carried out in future work to confirm whether the accuracy and F score can be maintained when using materials from various publishers.

Another area that should also be investigated is creating a model that allows multiple labels to be assigned to the same hierarchy and increases versatility for mathematical problems. In the experiment, we preprocessed the data to treat all digit expressions as the same, so in future work experiments should be carried out to check the influence of analyzing mathematical expressions on the classification. We also need to conduct another experiment to quantify the reduction in manual-labeling burden, as well as the difference in the rate of mislabelling from the perspective of the effect on educational consequences.

Moreover, the proposed method needs to address exercises that can be assigned multiple labels. Multi-label classification is also widely used in machine learning (Tsoumakas & Katakis, 2007). There are situations where multiple labels are given because one mathematical problem has multiple knowledge elements.

Acknowledgements

This work was partly supported by JSPS Grant-in-Aid for Scientific Research (B) JP20H01722 and JP23H01001, (Exploratory) JP21K19824, (Early Career) JP23K17012, (A) JP23H00505, and NEDO JPNP20006.

References

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

Chen, Q., Wang, W., Huang, K., & Coenen, F. (2022). Zero-shot text classification via knowledge graph embedding for social media data. *IEEE Internet of Things Journal, 9*(12), pp. 9205-9213.

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

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

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

Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 1*, pp. 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), pp. 349-359.

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

Gardner, M. W., & Dorling, S. R. (1998). Artificial neural networks (the multilayer perceptron) — a review of applications in the atmospheric sciences. *Atmospheric environment*, *32*(14-15), pp. 2627-2636.

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), pp. 4.

Ikeda, T. (2021). nagisa (0.2.7). https://github.com/taishi-i/nagisa.

Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou H., & Mikolov, T. (2017). Bag of tricks for efficient text classification. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, 2*, pp. 427-431.

Khan, M.F.F., Kanemaru, A., & Sakamura, K. (2022). Sentiment analysis of Japanese tweets using auto-augmented sentiment polarity dictionaries and advanced word embedding. *2022 IEEE 11th Global Conference on Consumer Electronics,* pp. 462-466.

Kobayashi, T. (2021). T-vMF similarity for regularizing intra-class feature distribution. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6616-6625.

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.

Palmer, J. A. (2021). pdftotext (2.2.2). https://github.com/jalan/pdftotext.

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), pp. 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), pp. 386.

Schubotz, M., Scharpf, P., Teschke, O., Kühnemund, A., Breitinger, C., & Gipp, B. (2020). Automsc: Automatic assignment of mathematics subject classification labels. *International Conference on Intelligent Computer Mathematics*, pp. 237-250.

Shen, J. T., Yamashita, M., Prihar, E., Heffernan, N., Wu, X., Graff, B., & Lee, D. (2021). MathBERT: A Pre-trained Language Model for General NLP Tasks in Mathematics Education. *NeurIPS 2021 Math AI for Education Workshop*.

Shen, J. T., Yamashita, M., Prihar, E., Heffernan, N., Wu, X., McGrew, S., & Lee, D. (2021b). Classifying math knowledge components via task-adaptive pre-trained bert. *International Conference on Artificial Intelligence in Education*, pp. 408-419.

Smith, R. (2007). An overview of the tesseract OCR engine. *Ninth International Conference on Document Analysis and Recognition,* pp. 629-633.

Sorower, M. S. (2010). A literature survey on algorithms for multi-label learning. *Oregon State University, 18*(1), pp. 25.

Takami, K., Dai, Y., Flanagan, B., & Hiroaki Ogata. (2022). Educational explainable recommender usage and its effectiveness in high school summer vacation assignment. *LAK22: 12th International Learning Analytics and Knowledge Conference.*

Takami, K., Flanagan, B., Dai, Y., & Ogata, H. (2021). Toward educational explainable recommender system: Explanation generation based on bayesian knowledge tracing parameters. *29th International Conference on Computers in Education Conference Proceedings, 2*, pp. 532–537.

Tian, Z., Flanagan, B., Dai, Y., & Ogata, H. (2022). Automated matching of exercises with knowledge components. *Proceedings of the 30th International Conference on Computers in Education Conference, 1*, pp. 23-31.

Tsoumakas, G., & Katakis, I. (2007). Multi-label classification: An overview. *International Journal of Data Warehousing and Mining, 3*(3), pp. 1-13.

Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of machine learning research, 9*(11), pp. 2579-2605.

Vapnik, V., & Lerner, A. (1963). Pattern recognition using generalized portrait method. *Automation and Remote Control, 24,* pp.774-780.

Vie, J. J., & Kashima, H. (2019). Knowledge tracing machines: Factorization machines for knowledge tracing. In: *Proceedings of the AAAI Conference on Artificial Intelligence, 33*(1), pp. 750-757.

Wang, J., Minematsu T., Okubo F., & Shimada. A. (2022). Topic-wise representation of learning activities for new learning pattern analysis. *30th International Conference on Computers in Education Conference, 1*, pp. 268-278.