IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 22, 2020, accepted October 23, 2020, date of publication November 20, 2020,
date of current version December 30, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3039531

A Self-Relevant CNN-SVM Model for Problem Classification in K-12 Question-Driven Learning

ERIC HSIAO-KUANG WU[1], (Member, IEEE), SUNG-EN CHEN[1], JHAO-JHONG LIU[1], YU-YEN OU[2], AND MIN-TE SUN[1], (Member, IEEE)
[1]Department of Computer Science and Information Engineering, National Central University, Taoyuan City 32001, Taiwan
[2]Department of Computer Science and Information Engineering, Yuan Ze University, Taoyuan City 32003, Taiwan

Corresponding author: Eric Hsiao-Kuang Wu (hsiao@csie.ncu.edu.tw)

ABSTRACT With the development and progress of science and technology, the learning patterns also evolve. In Question-Driven learning, students clarify and validate what they learn by answering questions. Such a large number of questions needs good management. A well-performed management can avoid the situation that learning materials with the same knowledge set are defined into different sections due to ambiguous expressions. In this work, we propose a hybrid classification model using the CNN-SVM that focuses on K-12 learning materials. We combine the Word2Vec feature and the hidden layer feature of CNN. In response to a current question that contains text and image, we also introduce a multi-modal preprocessing approach. The experiment results validate that the preprocessing method and the hybrid model can outperform the the state-of-the-art method and baseline methods.

INDEX TERMS Classification, convolutional neural network, support vector machine, Word2Vec, question-driven learning.

I. INTRODUCTION

In these days, formal education still remains as the core of learning while combining with new learning scenarios. There are multiple stages of education. The education before college is referred to as the K-12 system, including kindergarten, elementary school, middle school, and high school. In the K-12 system, the learner-centered learning method and means of self-regulated learning are effective learning types [1]. Students need to prepare for the exams and the large-scale competency tests that are used to assess learning. Thus, students often review and validate what they learn by answering questions. In our work, we focus on question-driven learning in the K-12 system.

In question-driven learning, a typical question contains only text, as shown in Fig. 1(a). However, sometimes text cannot clearly express the abstract information. Thus, adding images can make it much easier for students to understand the abstract information in the questions. For example, Fig. 1(b) is a question about calculating the area. When classifying the question, the geometric graph in Fig. 1(b) may play an important role. On the contrary, because the learning material is a piece of mathematical document, the text feature contains not only the narrative description of the content but also the mathematical expressions, which are also vital in question classification. Take Fig. 1(c) as an example, this question is an algebraic problem. However, if we only examine the narrative description, there is no knowledge point(labels) this question can be assigned to. In contrast, after incorporating the mathematical expressions, the question can be recognized as a polynomial related problem.

It is worth noting that the new learning materials come from various sources collected from Study123 Technology Co.Ltd in Taiwan simultaneously; the volume of the learning materials is increasing as time goes on and their content advances with time because of current affairs. Therefore, having a proper way to manage the self-relevant oriented K-12 learning materials becomes an issue that needs to be studied. Well-managed learning materials can help students understand and enhance the concept of knowledge. In most learning scenarios, keywords of knowledge are emphasized repeatedly. This phenomenon not only appears in the K-12 system, but also occurs in our lifelong learning journey. With this observation, managing learning materials by categorizing

The associate editor coordinating the review of this manuscript and approving it for publication was Hualong Yu.

If the area of a square is 20 square center meter and the perimeter is x center meter, then the value of x will between which two integers?

(A) 16, 17

(B) 17, 18

(C) 18, 19

(D) 19, 20

Answer: (B)

(a) A typical question that only contains narrative description.

As figure 1, In $\triangle ABC$, there are two points D 、E on the lines $\overline{AB}$ and $\overline{BC}$. If $\overline{AD}$ ： $\overline{DB}$ = $\overline{CE}$ ： $\overline{EB}$ =2 ： 3, then what is the proportion of the area of $\triangle DBE$ and $\triangle ADC$?

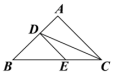
(A) 3 ： 5

(B) 4 ： 5

(C) 9 ： 10

(D) 15 ： 16

Answer: (C)



(b) A question on calculating area with a figure.

Calculate $6x \cdot (3-2x)$. Which of the following is the result of the function?

(A) $-12x^2+18x$

(B) $-12x^2+3$

(C) $16x$

(D) $6x$

Answer：(A)

(c) A question that mathematical expressions are the key of the classification

FIGURE 1. A typical question that only contains narrative description.

them into chapters and knowledge sets using the keywords of the knowledge can be an effective method.

There are several challenges of classifying learning material. First, to get better performance of classification, a good feature extractor and a good preprocess are necessary to make the features deliver the most category-specific information. Next, a powerful classifier is needed that is able to do best separation of the different categories in the feature space. Therefore, we apply a hybrid Convolutional Neural Network Support Vector Machine (CNN-SVM) Model [2], [3] for question-driven learning materials classification. The dense layer is replaced by SVM to get better global optimization in the feature space while convolutional layers are retained for feature extraction. Also, to manage the text, mathematical expressions, and images in our dataset, a series of preprocess methods is implemented. The proposed model can categorize mathematical learning materials in a question-driven learning scenario. Additionally, a retraining mechanism is designed so that the model can be strengthened simultaneously by its growing dataset. Thus, our model is self-relevant for learning material management. As the learning material dataset is growing, the proposed learning framework could be further updated for improved classification results.

II. RELATED WORK

In the past few years, the Machine Learning and Deep Learning technique have been applied to various fields, such as natural language processing, computer vision, bioinformatics, and artificial intelligence. The applications changed the world in many ways.

A typical way to manage learning materials is to use human labeling, but there are several drawbacks. First, owing to the data growth, the volume of learning materials can be considerable and ever growing. Second, human labeling may result in some bias label due to a personal subjective aspect. To address these issues, we can use a machine for classification and replace human labeling. A machine can reduce the human effort and eliminate the subjective aspect.

A. QUESTION CLASSIFICATION

Text classification is the ordinary way to preform a question classification problem. A recent work, Label-Embedding Attentive Model (LEAM) [11], has demonstrated the state-of-art performance in text classification. The fundamental difference between an ordinal text classification and K-12 math question classification is that K-12 math learning materials are both mathematical [6] and multi-media documents. As mathematical documents, math questions regularly include mathematical expressions. Moreover, some of the questions contain pictures of geometric figures to help the student understand the problems. On the contrary, the ordinal article to be classified may have more words in the documents and they do not contain any mathematical expressions.

B. MULTI-MODEL DOCUMENT

Many methods have been proposed for handling or classifying documents that contain different types of data. Some of the methods utilize relevant text information to enhance image features [7]. In [8], the authors introduce an approach that combines the textual and visual statistics as a single feature vector. The approach applies color histograms and a dominant orientation histogram to represent the images into vectors, then combines the vectors with a text vector generated from Latent Semantic Indexing. The authors prove that integrating features from images can produce a better result than the text-only model. The work also gives us the aspect of combining multi-modal data into a single vector. [9] present a model using a Bayesian network for classification of structured multimedia documents that contain text and images. The Bayesian network can model the relationship of each element in the document, thus classifying them with structural and content information. It utilizes TF-IDF for text and RGB histogram for image scoring. The above approaches are not suited for our dataset because our images are mostly geometric figures, which only consist of a white background and black lines. Hough transform is utilized in previous works, which are focused on extracting features or understanding graphs of mathematical questions [5]. As a result, we apply the concept from [5], that of using the Hough transform to extract the features from the diagram of geometry questions and apply Word2Vec for word embedding instead of TF-IDF. This gives us the similarity and relationship between words instead of pure statistical features.

C. HYBRID CNN-SVM MODEL

There are several hybrid models that combine the Convolutional Neural Network with the Support Vector Machine.

They are mostly used in computer vision tasks. [10] presents the model for invariant and perceptual texture mapping. The model contains a three-layer Convolutional Neural Network that connects to many Support Vector Machines. The result shows that it outperforms an individual neural network and SVM. [2] proposed a hybrid system that uses the Convolutional Neural Network to train and extract the feature, and a Gaussian-kernel Support Vector Machine is trained using the features from the Convolutional Neural Network. The model is used for generic object categorization, which is also a computer vision task. [3] combines the Convolutional Neural Network and the Support Vector Machine for visual pattern classification. The method uses the Convolutional Neural Network to learn features. Then it uses a Support Vector Machine to provide an optimal solution for the learned feature space. The proposed architecture is similar to the one in [2]. They both utilize the Convolutional Neural Network to learn and extract the feature from the dataset and apply the Support Vector Machine to find a hyperplane in the feature space.

III. SELF-RELEVANT CNN-SVM MODEL

The proposed model is separated into three parts: preprocess, document representation, and classifier. There are four processes in the preprocess part. First, a geometric features process transfers geometric features to keywords. Second, in mathematical expression process, patterns (such as regular expression) are used to capture commonly used formulas and then transferred to plain text keywords. We integrate those keywords from above processes with the noise eliminated plain text, and then do Chinese word segmentation as a tokenization process.

In the documentation representation part, Word2Vec is generated by using the tokenized text documents, and then CNN is used as a feature extractor. The hidden layer feature of CNN and the word vectors are combined as the text representation of the documents.

In the end, SVM is used as the classifier. The process of our classification system is depicted in Fig. 2

A. PART 1. PREPROCESSING

Our dataset, D, is composed of text (D_{text}) and images (D_{image}). First, the features in D_{image} are captured and represented in text. Then these features are integrated into D_{text}. After that, D is converted to a single modality dataset D'_{text} for classification.

1) IMAGE PREPROCESSING

The purpose of image preprocessing is to extract the geometric features from the images and convert them into keywords. In our dataset, since our learning materials images may contain geometric features, for example, composed of lines and points, it is not suitable to use the RGB histogram method to extract the features, as the images are mostly black and white.

In the image preprocessing part, the images are sent to the ConvNetJS convolutional neural network (CNN) model and

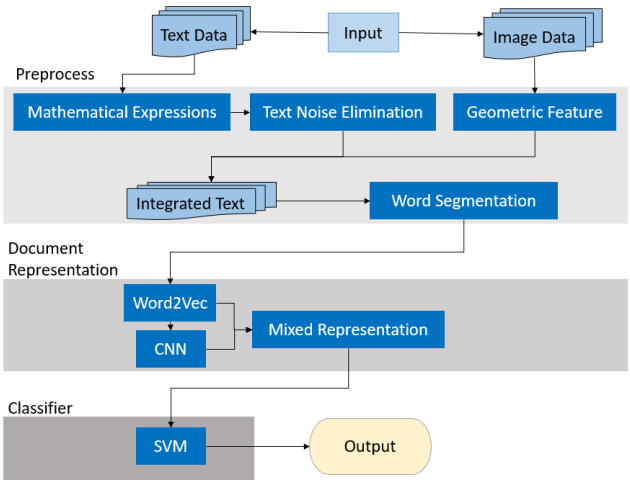


FIGURE 2. The processes of our classification system contain three stages: preprocessing, feature extraction and classifier.

the Hough transform. ConvNetJS is a Javascript library for training a deep learning model which is used to classify the images. The mathematical signs and fractions are images in the question set. These images are treated as noise, which is filtered to obtain a clear view of the question set.

After sieving about 600 images from 800 images with classification characteristics into 20 classes, we perform the image classification. The 200 removed images do not have knowledge feature, such as a symbol image. All of those images are transformed to 32 × 32 images. The network architecture is ConvNetJS CIFAR-10 Demo and the learning rate is set to be 0.00001 in our implementation. The network architecture is the same CNN in our hybrid model. After the classification is complete, we add the output of the model to the sample question.

First, each image in D_{image} is used to perform edge detection to get a clear contour. Then, the Hough transform is applied to the image from the data to extract the contour. The Scikit-image library provides built-in functions for us to utilize in our implementation. A set of rules R_{image} is defined for the composition of the extracted geometric features f_{image}. The processing of the image is denoted as $H(d_m, r_n)$, where m is the number of the images and n is the number of the rules.

$$f_{image} = \{H(d_i, r_i) \,|\, d_i \in D_{image}, r_i \in R_{image}\} \tag{1}$$

Fig. 3 shows two examples of our image preprocessing methods; we can infer that the first image contains a coordinate system by detecting the x-axis, the y-axis, and an origin O, which can be represented by the keyword "coordinate system". The second image contains a coordinate system and a straight line, which can be represented by keyword "linear equation".

The validation accuracy of the ConvNetJS CNN model is about 0.8; this means that some of the images may be misclassified. The keyword of the model output is not a crucial factor, because, as we shall see later, the Hough transform

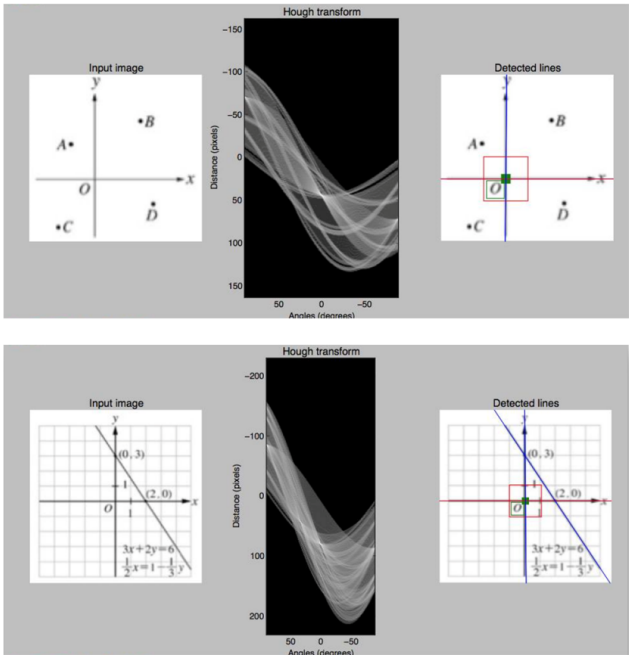


FIGURE 3. Two examples of image preprocessing after binarize and edge detection.

in the preprocessing part has excellent image detection accuracy. Moreover, the output of the text preprocessing part will improve the image classification. Therefore, the classification error resulted from CNN will only cause data noise rather than a detrimental consequence.

2) MATHEMATICAL EXPRESSIONS PREPROCESSING

As mentioned above, many of the mainlines in our learning materials are mathematical expressions. We observed that some category-specific formulas in learning materials have certain patterns since they are bounded by a limited knowledge set. Therefore, several regular expression patterns are used to parse the mathematical formulas before removing the punctuation. A different set of rules R_{RE} is defined for mathematical formulas for data in D_{text}. The documents are checked to see if there is a mathematical formula that matches any of the rules $r \in R_{RE}$. If a part of mathematical expression in $d_i \in D_{text}$ matches r, the matched expression will be replaced by keyword f_{RE}. The mathematical formula extracting process is represented as $M(d_m, r_n)$.

$$f_{RE} = \{M(d_i, r_i) | d_i \in D_{text}, r_i \in R_{RE}\} \tag{2}$$

3) TEXT NOISE ELIMINATION PREPROCESSING

This preprocess removes the noisy information in our dataset. For example, in Chinese characters, a number of punctuations does not provide additional information; therefore, they offer no contribution to our classification. Consequently, these punctuations are treated as noise and are removed from the dataset before the classification.

With this approach, our dataset is refined by removing noisy information.

4) TOKENIZATION

Tokenization is the process of cutting the corpus into a sequence of terms. The sequence of terms helps us learn the relationship from them. *Jieba* is the library applied in our tokenization process.

After text and image preprocessing, two modalities are merged into a single modality dataset D_{text}.

$$D'_{text} = f_{image} \cup D_{text} \cup f_{RE} \tag{3}$$

Jieba is performed to split our text corpus into a sequence of Chinese vocabulary. The tokenized D'_{texr} is denoted as T, where $T = \{d'_1, d'_1 \cdots, d'_m\}$. A document in T is d'_i, which consists of a sequence of tokens $(t_1, t_2, \cdots, t_n)$.

B. PART 2. DOCUMENT REPRESENTATION

Before classification, features need to be extracted from the preprocessed data and the feature vectors need to be generated for the classifier. Word2Vec and CNN feature extractor are applied in this part. Different from the traditional statistic methods of feature extraction, both of the applied models contain Neural Network, which allow us to extract the higher-level features by means of model training, such as the latent relationship between the word vectors. Our document representation concatenates the features of both Word2Vec and CNN. The following is a brief introduction of the techniques:

1) WORD2VEC

Word2Vec is a model for producing word embedding. It can learn the relationship between words. Moreover, Word2Vec can calculate similarities between words, and project them into vector space. The similarity can be evaluated via Euclidian distance or Cosine similarity. *Gensim* is the library utilized to generate Word2Vec. We apply continuous bag-of-words(CBOW) in our system because it is less time consuming. The input of the model is the tokenized data T from the previous processes. The learning materials have been tokenized to a sequence of Chinese vocabulary. The learning materials are used as the corpus and the dimension of word vectors is set to 250. The learning material corpus consists of a set of vocabulary, where vocabulary i is denoted as v_i. The dictionary of learning material corpus is denoted as $Dict_{lm} = \{v_1, v_2, \cdots, v_c\}$, where c is the number of all of the vocabulary in $Dict_{lm}$.

The input $(t_1, t_2, \cdots, t_n)$ is the sequences from T. After the training process, we obtain $Dict_{lm}$ and word vectors $W = \{V_1, V_2, \cdots, V_c\}$,where V_i is the word embedding of vocabulary v_i. To represent all tokens in $d'_i \in T$, a bag-of-word of W is applied to part of the representation of T. The Word2Vec representation of T, R_W, is a concatenated vector of V_i and zero padding z. Here, rw_i is a c×20−dimension document vector. By $Dict_{lm}$ vocabulary order, the wv_i in rw_i is V_i if v_i exists in d'_i; otherwise, wv_i is padded by z.

$$R_W = \{rw_1, rw_2, \cdots, rw_m\} \tag{4}$$

$$rw_i = \{wv_1, wv_2, \cdots, wv_c\} \tag{5}$$

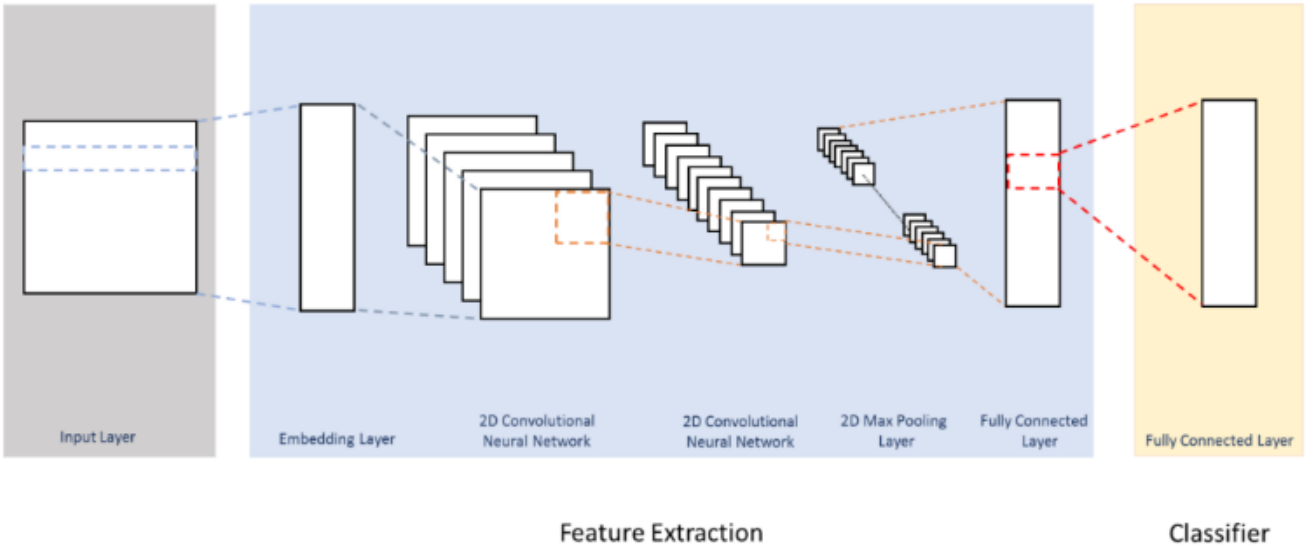


FIGURE 4. The architecture of our Convolutional Neural Network model.

$$wv_i = V_i, \quad if\ v_i \in d_i' \tag{6}$$
$$rw_i = z, \quad if\ v_i \notin d_i' \tag{7}$$

2) CONVOLUTIONAL NEURAL NETWORK

CNN is a multi-layer architecture that trains by the data and learns the higher-level features, such as local connectivity and special structures. The last layer will categorize the result. In this stage, we are going to extract the categorical local information of context. Our network architecture consists of an embedding layer, a couple of convolutional 2D layers, a max pooling layer, and a fully-connected layer. The architecture of our model is shown in Fig. 4.

At the beginning, the tokenized data T will be fed into the embedding layer, where W is set as the initial weights of the layer. While feeding the input data, the embedding layer will produce its corresponding word vector and move on to the next layer. The next layer consists of two Convolutional 2D layers. In both layers, the kernel size is set to 3×3 and the number of filters is set to 100. The decision of the parameters is made according to prior experiments. The kernel will extract the local feature from the input word vectors. After the Convolutional layers, a max pooling layer with pooling size 2×2 is added for down sampling. Max pooling layer passes the maximum value of the window to the next layer. After that, the network is flattened and connected with a fully-connected layer with 250 hidden nodes. This layer provides part of the document representation for our final classification. The last layer is a softmax layer, which is used to train and validate the network in the training process. After the training process of the CNN, the weight from the last fully-connected layer is extracted to generate the feature vector R_C as part of the input of the classifier.

The document representation R we input to classifier is the concatenated vector of R_W and R_C,where $R = \{r_1, r_2, \cdots, r_m\}$.

C. PART 3. CLASSIFICATION

After the features are extracted and the feature vector R is generated from the previous process, the document will be classified into categories. The classical concept of classification is as follows.

$$C = \{c_i\}_{i \in \{1,2,\cdots,|C|\}} \tag{8}$$
$$c_d = argmax_{c \in C} P(c|d) \tag{9}$$

C is the set of classes, and d is the true class of the data. SVM is utilized as our classification model, which is implemented by importing the Sci-kit Learn library.

Given a set of training data with m points of the form,

$$(r_1, y_1), \ldots, (r_m, y_m) \tag{10}$$

where r_i is a vector that represents the data, and y indicates the class that r_i belongs to. The goal of SVM is to find a hyper-plane that separates the data into the classes where they belong.

D. RETRAINING MECHANISM

Due to the ever growing data from the volume of learning materials, the classification model needs to update with the latest dataset to maintain the best-suited classification. Our model provides a retraining mechanism to keep the classifier up to date. Generally, the input questions will be classified by the trained model. After the new classified learning materials reach a certain quantity, the model should automatically initiate the retraining mechanism and our model will be tuned by the new data. A retraining mechanism is implemented to achieve this goal. First, a threshold is defined to determine if the retraining is necessary. After the size of new input data exceeds the threshold, the retraining mechanism will automatically be triggered and initiate the training process of the model with the latest dataset. However, the classifier will be retrained completely by all of the updated data, because

SVM cannot be partially retrained. Through this retraining mechanism, the model could constantly improve with the growing dataset.

Algorithm 1 Algorithm of the Retraining Mechanism

Require:
New learning materials for classification, D_{new};
Retraining threshold, $Thres_{retrain}$;
Original trained model, $M_{original}$;
Ensure:
New model after retraining M_{new};
Predict D_{new} with $M_{original}$ and count the input I;
if $I > Thres_{retrain}$ **then**
    retrain the model with all data and get M_{new};
end if

IV. EVALUATION

Our dataset contains 1400 mathematical questions and some of them consist of an image; the total number of images is 600. There are 13 categories of these mathematical questions. Each result is obtained by running the experiment 10 times and computing the average of them. To test the performance boost of the multi-modal preprocessing methods, the classification result is tested starting from text only, gradually adding the mathematical expression feature extraction and then the image features integrating with text.

A. DATA SETTING

The dataset is divided into two parts: 80 percent for training, and 20 percent for testing. In order to avoid overfitting of our CNN model, in CNN stage, we split the training data into two parts: 10 percent for validation and 90 percent for training. After that, the training data is fed into the SVM to train the classifier. After the training process, the testing data in our trained model are used to evaluate the performance of our model.

B. HYPERPARAMETERS

We are interested in identifying the combination of parameters that results in the best performance of our model. Through experiments, we have found that parameters like dropout rate, embedding dropout rate, hidden dimension, and embedding dimension affect the performance the most. For other parameters such as the batch size and learning rate, they are set to be 32 and 0.0001 by default while training. First, we conduct a grid search on the embedding dimension and hidden layer dimension, and the results are shown in Fig. 5 and Fig. 6. As shown in the figures, the embedding dimension and hidden layer dimension seem to perform the best in the range of 250 and 300, respectively. To identify the best combination of the embedding dimension and hidden layer dimension, we set the embedding dimension and hidden layer dimension to be in the range of 150 and 350 and in the range of 200 to 400, respectively, for a double-parameter grid search, as shown in Fig. 9. As can be observed in the figure, the best combination of the embedding dimension and hidden layer dimension is (300, 250). In addition, another double-parameter grid search is conducted on CNN dropout rate and embedding dropout rate. In the grid search, the CNN and embedding dropout rates are centered at 0.5 and 0.3, respectively, based on the Keras official document default values [12]. The results of the grid search are shown in Fig. 8. As can be seen in the figure, the best combination of the CNN dropout and embedding dropout rates is (0.5, 0.3) with rbf kernel and Adam optimizer.

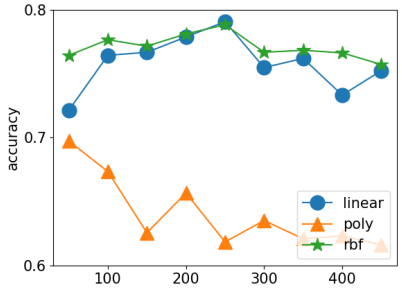


FIGURE 5. Accuracy vs. embedding dimension.

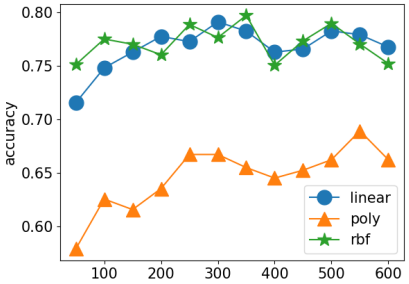


FIGURE 6. Accuracy vs. hidden layer dimension.

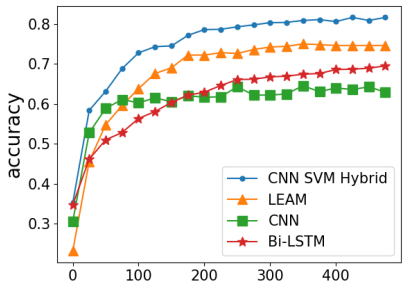


FIGURE 7. Comparison of different method.

C. BASELINE METHODS

Our model will be compared with the state-of-the-art model, LEAM [11]. LEAM demonstrates text classification using both label embedding and word embedding to get the word-label attention for text classification. In addition, we are also interested in how the classic text classification model performs in our question classification problem. Therefore, we also include CNN and Bi-LSTM as our baseline methods. The CNN classifier has the same network topology as our CNN feature extraction module. LSTM is a well-known

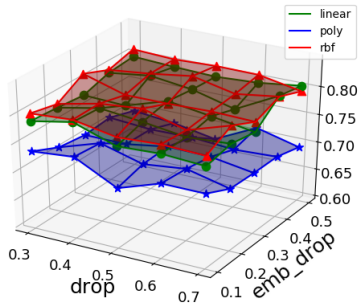


FIGURE 8. Accuracy vs. CNN dropout rate vs. embedding dropout rate.

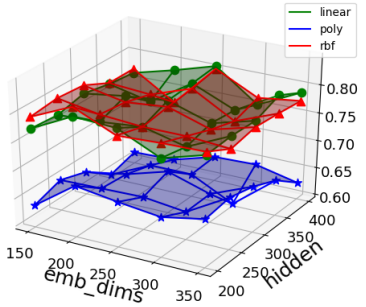


FIGURE 9. Accuracy vs. hidden layer dimension vs. embedding dimension.

neural network in natural language processing field [4]. The Bi-LSTM utilizes bi-directional LSTM. In the Bi-LSTM classifier, the number of Bi-LSTM cells is set to be 50 with a 1D max pooling layer and a dense layer for classification. Both the CNN and Bi-LSTM have an embedding layer after the word sequence input T.

D. EVALUATION METRIC

We evaluate the classifier's performance by calculating the accuracy score (notations are defined in Table 1).

$$Accuracy = \frac{TP + TN}{N}, \tag{11}$$

where ''true'' means that the document is predicted correctly, positive means that the document is predicted to belong to positive class, and vice versa.

E. EXPERIMENT RESULT

In Table 2, it can be seen that the CNN-SVM hybrid model outperforms the state-of-art text clasification method, LEAM [11]. On the contrary, it can be seen that the CNN-SVM hybrid model outperforms Bi-LSTM by almost 10% and CNN by 15% in terms of accuracy. This indicates that our method is more suitable for the question classification dataset. The SVM is good at finding general optimal solution for the learned Word2Vec feature space. Combining SVM with CNN can take advantages from both of the models.

Table 3 shows the performance of the CNN-SVM hybrid model with different feature processing methods. The experiment result shows that text is the most important feature in our model, and the mathematical expression features help our model perform better. It can be seen that our image processing method does not provide significant performance improvement in our experiment. This is due to the fact that keywords obtained from those images in the learning materials are likely to appear in the text description of the questions. Additionally, the images that contain general mathematical meanings are not that common in the K-12 learning materials, as many images merely illustrate the scenario of the questions.

TABLE 1. Definition of notation.

Symbol	Meaning
D	Dataset
D_{text}	Dataset of text
D_{image}	Dataset of image
D'_{text}	Dataset that integrated with image features
R_{image}	Rules that are defined for the composition of the extracted geometric features
f_{image}	The keywords of extracted geometric features
R_{RE}	Rules that are defined for extracting mathematical formulas
f_{RE}	The mathematical formulas keywords
T	The tokenized D_{text}
d'_i	The i th document in T
$Dict_{lm}$	The dictionary of learning material corpus
v_i	The ith vocabulary of $Dict_{lm}$
c	The number of vocabulary in the learning material corpus
W	Word vectors of the learning material corpus
V_i	he word vector of v_i
z	Zero padding vector
R_W	The part of document representation of Word2Vec
R_C	The part of document representation of CNN
R	The document representation
C	The set of classes
TP	The number of documents that is true positive
TN	The number of documents that is true negative
N	The number of documents

TABLE 2. Accuracy of the methods.

Classification Method	Accuracy
Hybrid CNN-SVM	0.837
CNN	0.685
Bi-LSTM	0.730
LEAM[11]	0.770

TABLE 3. Comparison of preprocessing Methods.

Preprocess Method	Accuracy
Text + ME + Image	0.837
Text + ME	0.827
Text	0.834

After checking the misclassified documents, most of the mistakes happen on the labels that have high correlations, such as two sub-concepts of single variable algebra, or question of proportional with two variables or multiple variables. In addition, many misclassified documents are recognized as ambiguous between the predict label and the true label. Moreover, in some of the misclassified documents the predict labels are actually the correct ones because the original labels are incorrect. Hence, our method correctly classifies the document most of the time as long as the document is not ambiguous among multiple labels with high correlation.

The above experiments prove that combination of the multi-modal preprocessing, Word2Vec, and the hybrid model can improve the performance of the classification. Our model can reduce the labelling human effort and eliminate the subjective mistakes.

V. CONCLUSION

In this work, we introduce a hybrid CNN-SVM for classification of the mathematical learning materials in question-driven learning scenarios. Our proposed model can handle learning materials with multiple modalities such as text and images simultaneously. Furthermore, we extract the mathematical formulas in text as the features for classification. The result shows that though combining multiple modalities does not improve the performance of the classification, our feature extraction method of mathematical expression does improve the performance of the classification.

Our model can accurately classify the data into chapter and knowledge sets. The experiments show that our hybrid CNN-SVM model is effective, and it outperforms the CNN and Bi-LSTM model. The model can also deal with the cross-topic labeling situation and retrain itself along with the data growth.

VI. FUTURE WORK

We believe that in the future, this classification model can be utilized in other subjects, such as English, science, and language questions. Moreover, by integrating it with student learning portfolios, the model can be extended to a recommendation system for learning. It can analyze the students' learning process, and provide feedback for students. The system can track the weaknesses of the students and supply them with the suitable learning materials to improve their learning performance. The recommendation system can turn into a customized personal learning assistant that is specialized for every unique student.

This system can give help to not only to students but also to instructors. Instructors can manage and modify the lessons using the collected data from the students. For example, it offers a more efficient way to schedule the content of the quiz if the system can recommend learning materials along with the given knowledge set from the instructors.

We can further apply semantic analysis to the learning materials. Since the machine understands the questions, it can provide corresponding answers to the questions. Hence, we can extend the classification model in a question-answering system for K-12 students. With the combination of the recommendation system and the question-answering system, we believe that it can offer significant assistance to students for learning progress.

REFERENCES

[1] S. J. H. Yang, J. C. H. Huang, A. Y. Q. Huang, and I. Y. L. Chen, "MOOCs for K-12 and higher education in taiwan," in *Proc. 5th IIAI Int. Congr. Adv. Appl. Informat. (IIAI-AAI)*, Jul. 2016, pp. 361–365.

[2] J. Nagi, G. A. D. Caro, A. Giusti, F. Nagi, and L. M. Gambardella, "Convolutional neural support vector machines: Hybrid visual pattern classifiers for multi-robot systems," in *Proc. 11th Int. Conf. Mach. Learn. Appl.*, Dec. 2012, pp. 27–32.

[3] X.-X. Niu and C. Y. Suen, "A novel hybrid CNN–SVM classifier for recognizing handwritten digits," *Pattern Recognit.*, vol. 45, no. 4, pp. 1318–1325, Apr. 2012.

[4] P. Zhou, Z. Qi, S. Zheng, J. Xu, H. Bao, and B. Xu, "Text classification improved by integrating bidirectional LSTM with two-dimensional max pooling," 2016, *arXiv:1611.06639*. [Online]. Available: http://arxiv.org/abs/1611.06639

[5] M. Dalal and M. A. Zaveri, "Automatic text classiï cation: A technical review.," *Int. J. Comput. Appl.*, vol. 28, no. 2, pp. 37–40, 2011.

[6] T. Suzuki and A. Fujii, "Mathematical document categorization with structure of mathematical expressions," in *Proc. ACM/IEEE Joint Conf. Digit. Libraries (JCDL)*, 2017, pp. 1–10.

[7] M. La Cascia, S. Sethi, and S. Sclaroff, "Combining textual and visual cues for content-based image retrieval on the world wide Web," in *Proc. IEEE Workshop Content-Based Access Image Video Lib.*, 1998, pp. 24–28.

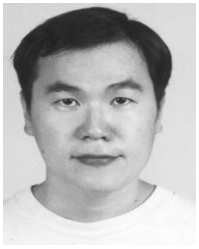
[8] L. Denoyer, J.-N. Vittaut, P. Gallinari, S. Brunes-Saux, and S. Brunessaux, "Structured multimedia document classification," in *Proc. ACM Symp. Document Eng.*, 2003, pp. 153–160.

[9] M. N. Seo, "Diagram understanding in geometry questions," in *Proc. AAAI*, 2014, pp. 1–5.

[10] F. Jie Huang and Y. LeCun, "Large-scale learning with SVM and convolutional for generic object categorization," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 284–291.

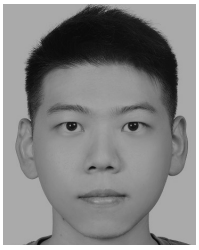
[11] G. Wang, C. Li, W. Wang, Y. Zhang, D. Shen, X. Zhang, R. Henao, and L. Carin, "Joint embedding of words and labels for text classification," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, 2018, p. 2321.

[12] F. Chollet *et al. Keras*. Accessed: 2015. [Online]. Available: https://keras.io



ERIC HSIAO-KUANG WU (Member, IEEE) received the B.S. degree in computer science and information engineering from National Taiwan University, in 1989, and the master's and Ph.D. degrees in computer science from the University of California, Los Angeles (UCLA), in 1993 and 1997, respectively. He is currently a Professor of computer science and information engineering with National Central University, Taiwan. His research interests include wireless networks, mobile computing, and broadband networks. He is also a member of the Institute of Information and Computing Machinery (IICM).

SUNG-EN CHEN received the B.S. and M.S. degrees in computer science and information engineering from National Central University, Taoyuan City, Taiwan, in 2017 and 2019, respectively. Her research interests include machine learning and data mining.



JHAO-JHONG LIU received the B.S. degree in computer science and information engineering from National Chi Nan University, in 2019. He is currently pursuing the M.S. degree with the Department of Computer Science and Information Engineering, National Central University, under the supervision of Prof. E. H.-K. Wu. His research interests include machine learning and image processing.

YU-YEN OU is currently an Assistant Professor with the Department of Computer Science and Engineering, Yuan Ze University, Taiwan. His research interests include machine learning and bioinformatics.

MIN-TE SUN (Member, IEEE) received the B.Sc. degree from National Taiwan University, the M.Sc. degree from Indiana University, Bloomington, and the Ph.D. degree in computer and information science from The Ohio State University. He is currently a Professor with the Department of Computer Science and Information Engineering, National Central University, Taiwan. His research interests include distributed computing and the IoT. He is also a member of the ACM.