Coding Collaboration Process Automatically: Coding Methods Using Deep Learning Technology

Kimihiko Ando
Cloud Service Center
Tokyo University of Technology
Tokyo, Japan
email:ando@stf.teu.ac.jp

Chihiro Shibata
School of Computer Sciences
Tokyo University of Technology
Tokyo, Japan
email:shibatachh@stf.teu.ac.jp

Taketoshi Inaba
Graduate School of Bionics, Computer and Media Sciences
Tokyo University of Technology
Tokyo, Japan
email:inaba@stf.teu.ac.jp

*Abstract***— In Computer Supported Collaborative Learning (CSCL) research, gaining a guideline to carry out appropriate scaffolding by analyzing mechanism of successful collaborative interaction and extracting indicators to identify groups where collaborative process is not going well, can be considered as the most important preoccupation, both for research and for educational implementation. And to study this collaborative learning process, different approaches have been tried. In this paper, we opt for the verbal data analysis; the advantage of this method is that it enables quantitative processing while maintaining qualitative perspective, with collaborative learning data of considerable size. However, coding large scale educational data is extremely time consuming and sometimes goes beyond men's capacity. So, in recent years, there have also been attempts to automate complex coding by using machine learning technology. In this background, with large scale data generated in our CSCL system, we have tried to implement automation of high precision coding utilizing deep learning methods, which are derived from the leading edge technology of machine learning. The results indicate that our approach with deep learning methods is promising, outperforming the machine learning baseline. But the prediction accuracy could be improved by constructing coding schemes and models more sensitive to the context of collaboration and conversation. Therefore, we propose a new coding scheme that can represent the context of learning more comprehensively and accurately at the end of this paper for the next research.**

Keywords-CSCL; leaning analytics; coding scheme; deep learning methods.

I. INTRODUCTION

This article is an extended version of a conference paper presented at eLmL 2017, the Ninth International Conference on Mobile, Hybrid and On-line Learning [1]. It introduces more information on the theoretical background of this study and especially a new coding scheme, based on the experiment results.

A. Analysis of collaborative process

One of the greatest research interests in the actual Computer Supported Collaborative Learning (CSCL) research is to analyze its social process from a social constructionist viewpoint, and key research questions are as follows: how knowledge and meanings are shared within a group, what types of conflict, synchronization and adjustment of opinions occur, and how knowledge is constructed from discussions. And answering to these questions enables to develop more effective scaffolding methods and CSCL system and tools.

In earlier researches at initial stage of CSCL, the focus was on each individual within a collaborating group, and the main point of interest had been how significantly a personal learning outcome was affected by characteristic types of a group (such as group size, group composition, learning tasks, and communication media) [2]. However, it gradually became clear that those characteristics are complexly connected and intertwined with each other, and showing causal relation to a specific result was extremely difficult. From the 1990s, the interest in CSCL research had moved away from awareness of the issue on how a personal learning is established within a group, to attempting to explain the process by clarifying the details of group interactions when learning is taking place within a group [3].

However, attempting to analyze collaborative process goes beyond merely shifting a research perspective; it also leads to fundamental re-examination of its analytical methodology. In other words, this involves a shift from quantitative analysis to qualitative analysis. Naturally, there are useful data among quantitative data saved within CSCL system, such as the number of contributions within a group, the number of contributions by each group member, and in some cases contribution attributes obtained from system interface (sentence opener), but those are very much a mere surface data. The most important data for analysis are contributions in chats, images/sounds within tools such as Skype, and various outputs generated in the process of collaborative learning; for analysis of those, ethnomethodologies such as conversation analysis and video analysis have been invoked [4][5].

However, those researches by their very nature tend to be in-depth case studies of collaborative activities with a limited number of groups and have the disadvantage of not at all being easy to derive a guideline that has a certain level of universality and can be applicable in other contexts. Therefore, researches have been carried out using verbal data analysis method that carry out coding from a perspective of linguistic or collaborative learning activities on a certain volume of language data generated in collaborative learning and analyzing them [6][7][8]. The advantage of this method is that it enables quantitative processing while maintaining qualitative perspective, with collaborative learning data of considerable size as the subject, while coding them manually is an extremely time consuming task, which goes sometimes beyond men's capacity. For example, Persico et al. developed a technological tool which helps the tutors to code the contributions in chats and displays quantitative information about the qualitative information and coding data [9]. However, given that the coding procedure itself remains manual in most existing studies [10][11], there is an insurmountable limit in front of big data. Hence, we seek an automatic coding technique for a large scale collaborative learning data with deep learning methods.

B. Educational data and Learning Analytics

With the progress of educational cloud implementation in educational institutions, data generated in Learning Management System (LMS), e-learning, Social Network Service (SNS), Massive Open Online Course (MOOC) and others are increasing rapidly, and a new research approach called Learning Analytics (LA) that tries to gain knowledge that would lead to support of learning and educational activities by analyzing those educational big data is becoming more active [12][13]. Big educational data obtained from CSCL system integrated in educational cloud at a campus, such as conversation data, submitted documents and images/sounds of learning activities, will certainly become a subject for analysis in the near future: therefore, it is believed that we are coming into a time when it is necessary to seriously examine a new possibility of collaborative learning research as LA. Due to such background, in this research we have reconstructed CSCL system that has been operating in a campus server for the last five years as a module within Moodle, which is a LMS within the campus cloud, and have already structured an environment that can be operated within the campus and collect/analyze collaborative learning data.

C. The goal and purpose of this study

The goal of our research is to analyze large-scale collaborative data from the perspective of LA as described above and discover the mechanism of activation and deactivation of collaborative activity process which could not be gained from micro level case studies up to now. Furthermore, this research, based on its results, aims to implement supports in authentic learning/educational contexts, such as real-time monitoring of collaborative process and scaffolding to groups that are not becoming activated.

In this paper, as the first step towards this goal, we present work in progress, which attempts to develop an automation technique for coding of chat data and verifies its accuracy. To be more specific, a substantial volume of chat data is coded manually, and has a part of that learnt as training data in deep learning methods, which are derived from the leading edge technologies for machine learning; afterwards, automatic coding of the raw data is carried out. For validation of accuracy, the effectiveness of using deep learning methods is assessed by comparing accuracy against Naive Bayes and Support Vector Machines, which are baselines of machine learning algorithm used in existing studies that carried out automatic coding by machine learning.

D. Structure of this paper

This paper is structured as follows. In Section II, we present the related work. The Section III describes our datasets and coding scheme. The approach with deep learning methods for automatic coding is discussed in Section IV. Then, our experiment and results from our evaluation are described in Section V. In Section VI, taking account of experimental results, we propose a new coding scheme. Section VI concludes the paper.

II. RELATED WORK

Since deep learning can often outperform existing machine learning methods, such as SVMs, it has been applied in various research areas, such as image recognition and natural language processing [14]. Text classification is an important task in natural learning processing, for which various deep learning methods have been exploited extensively in recent studies. A structure called a CNN has been applied for text classification using word- or character-level modeling [15][16]. LSTM [17] and gated recurrent units (GRUs) [18] are popular structures for RNNs. Both structures are known to outperform existing models, such as n-grams, and thus are widely available as learning models for sequential data like text. RNNs are also applied to text classification in various ways [19][20]. For instance, Yang et al. used a bidirectional GRU with attention modeling by setting two hierarchical layers that consist of the word and sentence encoders [19].

In the field of CSCL, some researchers have tried to apply text classification technology to chat logs. The most representative studies would be Rosé and her colleagues' works [21][22][23]. For example, they applied text classification technology to a relatively large CSCL corpus that had been coded by human coders using the coding scheme with multiple dimensions, developed by Weinbeger and Fisher [22][24]. McLaren's Argunaut project took a similar approach: he used online discussions coded manually to train machine-learning classifiers in order to predict the appearance of these discussions characteristics in the new e-discussion [25]. However, it should be pointed out that all these prior studies rely on the machine learning techniques before deep learning studies emerge.

III. DATA AND CODING SCHEME

In this section, we explain how we collected our dataset and what coding scheme we adopted to categorize the dataset.

A. *Data Description*

Our dataset obtained through chat function within the system, comes from conversations among students while carrying out online collaborative learning in university lectures using CSCL, which had been previously developed by the researchers of this study [26].

This CSCL is used without face to face contact; therefore, these data are all from occasions when unacquainted and separated students formed groups within lecture halls at the campus. And within the system all names of students are shown in nicknames, so that even if students knew each other they would not recognize each other.

The overview of CSCL contributions data used in this research is shown in Table I. The number of lectures is seven and all classes of these lectures form groups of three to four; in fact, there are a lot of data that we could not process by coding them in this research. Learning times vary depending on the class, from 45 to 90 minutes. In total, the dataset contains 11504 contributions; there are 202 groups from all the classes, with 426 participating students; since students attend multiple classes, the number of participating students are smaller than the product of number of groups and number of students in a group.

Table II shows a conversation example of chat. This is a conversation example of three students.

TABLE I. CONTRIBUTIONS DATA USED IN THIS STUDY

Number of Lectures	7 Lectures
Member of Groups	3-4 people
Learning Time	45-90 mintutes
Number of Groups	202 groups
Number of Students	426 students

TABLE II. CONVERSATION EXAMPLE (TRANLATION FROM JAPANESE)

Talker	Contents
D	Where do you want to change?
E	That's right … I guess, first of all, we definitely need to change the question, and then, what about the well-formed formula?
D	How is it that changes only the third line of the question?
D	Regarding the well-formed formula, it's the final part after ⊃.
E	That's good idea.
F	I agree. How do we want to change that?

B. *Coding scheme*

In accordance with our manual for code assignment, one code label is assigned to one contribution in a chat. There are 16 types of code labels as shown in Table III, and one of those labels is assigned for all cases.

All labels in our dataset are coded by two people; the coincidence rate between the labels assigned was 67%. However, when we reviewed the resultant coding data, it was discovered that there were duplicated labels for some contributions, and some labels had variances depending on the coder; therefore, after conferring among us, we unified labels and re-coded the contributions. The resultant number of labels assigned is shown in Table III. Concordance rate is 82.3% and this is a high concordance rate with 0.800 Kappa coefficient, and we consider this to be sufficiently practical for use as an educational dataset in deep learning methods. Fig. 1 shows the frequencies of the labels in the dataset. Nine labels describe more than 90% of occurrences; label occurrences appear to have a long-tail distribution. The main purpose of this study is to learn and infer these labels from posted contributions.

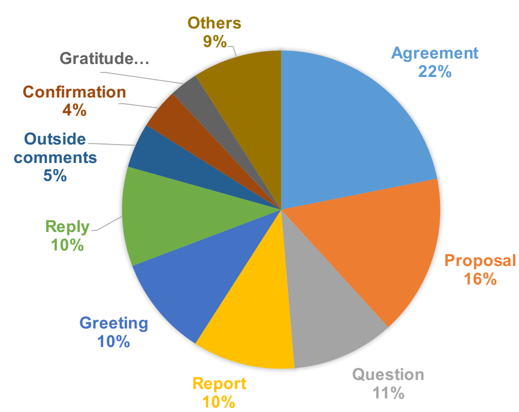

Figure 1. Ratio of each conversational coding labels

IV. APPROACH –DEEP LEARNING

In recent years, deep learning technology has led to dramatic developments in the field of artificial intelligence. Deep learning is a general framework of learning methods that use neural networks with millions of weight parameters. The weights in neural networks are optimized so that their output coincides with labels in the given data. With the recent development of parallel computing using Graphics Processing Units (GPUs) and optimization algorithms, machines are able to learn large numbers of parameters from large datasets at realistic costs.

To try automatic coding, we adapt three types of deep neural network (DNN) structures: a convolutional neural network (CNN) based model and two bidirectional Long short-term memory (LSTM) based models, LSTM and Sequence-to-Sequence (Seq2Seq). The first and second models take only a single contribution as input and cannot refer to context information in the conversation. Conversely, the Seq2Seq model can capture context information by using

TABLE III. List of labels

Label	Meaning of label	Contribution example	Number of times used
Agreement	Affirmative reply	I think that's good	5033
Proposal	Conveying opinion, or yes/no question	How about five of us here make the submission?	3762
Question	Other than yes/no question	What shall we do with the title?	2399
Report	Reporting own status	I corrected the complicated one	2394
Greeting	Greeting to other members	I'm looking forward to working with you	2342
Reply	Other replies	It looks that way!	2324
Outside comments	Contribution on matters other than assignment contents	My contribution is disappearing already; so fast!	1049
	Opinions on systems and such	A bug	
Confirmation	Confirm the assignment and how to proceed	Would you like to submit it now?	949
Gratitude	Gratitude to other members	Thanks!	671
Switchover	A contribution to change event being handled, such as moving on to the next assignment	Shall we give it a try?	625
Joke	Joke to other members	You should, like, learn it physically? :)	433
Request	Requesting somebody to do some task	Can either of you reply?	354
Correction	Correcting past contribution	Sorry, I meant children	204
Disagreement	Negative reply	I think 30 minute is too long	160
Complaint	Dissatisfactions towards assignments or systems	I must say the theme isn't great	155
Noise	Contribution that does not make sense	?meet? day???	143

a pair of sentences as its input, which represent source and replay contributions.

A. CNN-based model

The CNN-based model uses the network architecture proposed by Kim et al. (Fig. 2). Before training, all words in the data are converted to word vectors. Word vectors are often obtained by pre-training using another external dataset. In this study, we implemented two types of word vectors: 1) vectors obtained by applying word2vec (the skipped gram model with negative sampling) to all Japanese text in Wikipedia, and 2) randomly initialized vectors that are tuned simultaneously with the CNN.

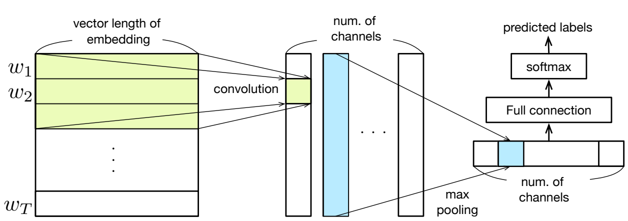

Figure 2. CNN-based model

B. Bidirectional LSTM-based model

An LSTM is a recurrent neural networks (RNNs) that is carefully constructed so that it can capture long-distance dependencies in sequential data. Generally speaking, an RNN consists of input vector x_t and output vector y_t for each time t. To obtain the output $y_{\{t\}}$, the previous output vector $y_{\{t-1\}}$ is fed to the neural network along with the current input vector x_t. The LSTM has another hidden vector, c_t, called the *state vector* in addition to the input and output vectors. While the state vector is also output from the neural network, it is computed to track long-distance relations through a function called a *forget gate,* which is designed to decide whether the state vector should be changed. We feed word vectors into the two-layer LSTM network sequentially in both the forward and reverse directions. After all words in a contribution are input, both output vectors are concatenated and fed into the two-layer fully-connected network and the softmax layer to obtain classification results. Fig. 3 illustrates this architecture.

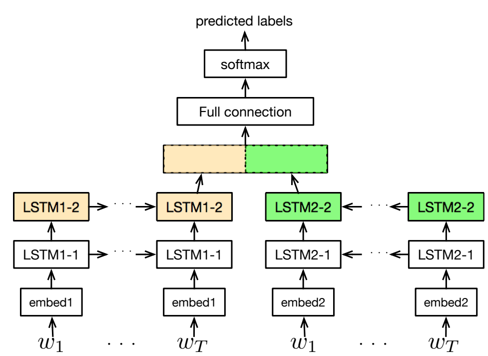

Figure 3. Bidirectional LSTM-based

C. Bidirectional Seq2Seq-based model

Each contribution is a part of a conversation; therefore, to classify labels more accurately, we must account for conversational contexts. To do this, we convert all contributions in conversations into pairs of *source* and *reply* contributions (Table IV). Even if a user posts a contribution that does not explicitly cite another, we assume that it cites a previous contribution. We also suppose that the first contribution of each conversation cites the empty string. To construct a model that regards the source contribution as a conversational context and the reply as a representation of the user's intention, we use the Seq2seq framework. Seq2seq [27] was originally proposed as a neural model using RNNs for machine translation, and later applied to other tasks, such as conversational generation [28]. It consists of two separate LSTM networks, called the encoder and decoder. We use two-layer LSTM networks for both the encoder and decoder. Words are sequentially fed in both the forward and reverse directions. Output vectors from decoders are concatenated and fed into the two-layer fully-connected network and the softmax layer (Fig. 4).

TABLE IV. Examples for source and replay contributions

Source (u)	Replay (w)	Label
(None)	How about five of us here make the submission?	**Proposal**
(None)	I must say the theme isn't great.	**Com-plaint**
How about five of us here make the submission?	It sounds great!	**Reply**
I must say the theme isn't great.	If we had another hour, we could change it…	**Agree-ment**
It sounds great!	Thanks!	**Gratitude**

V. EVALUATION

A. Data Preprocessing

For each contribution, we trimmed sentences beginning with the symbol ">," which were automatically generated by the system. Since all the data consist of Japanese text, morphological analysis was needed. We split texts into words using a tool called MeCab [29]. Replacing low-frequency words with "unknown," the vocabulary size was decreased to approximately 4,000. Each contribution was given two labels annotated by different people; we removed contributions that were assigned two different labels. We used 90% of the remaining 8,015 contributions as training data and 10% as test data. The accuracy of the learning result for each model is measured with the test data.

B. Baseline Methods

For comparison, we used three classifiers; Naive Bayes, a linear support vector machine (SVM), and an SVM with a radial basis function (RBF) kernel. We also used two types of feature sets: unigrams only and unigrams and bigrams. For the SVM classifiers, in order to improve the classification accuracy, input vectors were obtained by normalizing zero-one vectors whose elements represent occurrences of unigrams or bigrams.

C. Model Parameters and Learning

Model parameters, such as the vector sizes of layers, are determined as follows. Both the size of word embedding and the size of the last fully connected layer are 200 for all models. We set the patch size of the convolutional layer in the vertical direction to 4 and the number of channels to 256 for the CNN-based models. We set the size of both LSTM layers to 800 for the LSTM and Seq2Seq models. The set of parameters were needed to be chosen so that their prediction accuracy of the model will not be reduced, and at the same time, the computational cost of learning is in the range of reasonable time. Generally, the vector size of LSTM layers is needed to be increased for better prediction accuracy when it is inappropriately small. On the other hand, if it is sufficiently large, increasing their size is almost in vain for better accuracy. For instance, if we set it larger than that of our setting, say 1000 or 2000, we will get almost the same value of accuracy as the result of the experiment. Thus, we empirically decided it so as to achieve the nearly optimal accuracy and to minimize computational cost. Meanwhile, we need to carefully choose the vector size of the last fully connected layer. Our model easily suffers from over fitting if we set it too large. On the other hand, if we set it too small, our model is suffered from the lack of the expression capability. Thus, we should set it moderately; not so small to

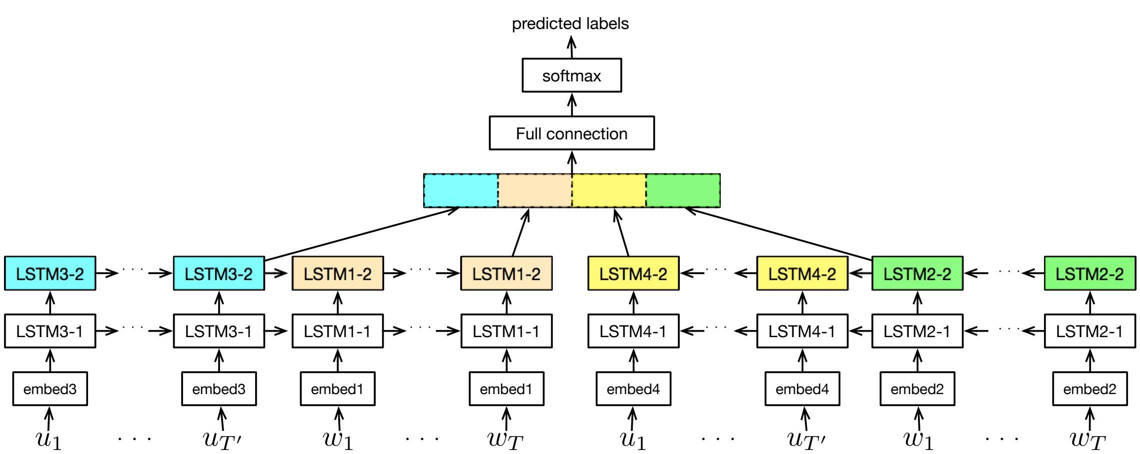

Figure 4. Bidirectional Seq2Seq-based model

have the sufficient capability to learn accurately, and not so large to avoid the over fitting problem. We obtained 200 as an appropriate value for the vector size of the last layer through several experiments.

Models are learned by stochastic descent gradient (SDG) using an optimization method called Adam. To avoid overfitting, iteration was stopped at 10 epochs for the LSTM-based methods and 30 epochs for the CNN-based methods. Due to the fluctuation in accuracy results between epochs, we took the average of the last 5 epochs to measure the accuracy of each model. To prevent overfitting, dropout was applied to the last and second-last fully connected layers. Figure 5 shows the learning curves of the CNN-based model with Wikipedia and the bi-directional Seq2Seq-based model. The y-axis shows the accuracy on the test data. As the figure shows, the accuracy converges approximately after around 10 epochs for the Seq2Seq-based model. On the other hand, it converges after around 30 epochs. The numbers of epochs that are needed for convergence largely depend on the models.

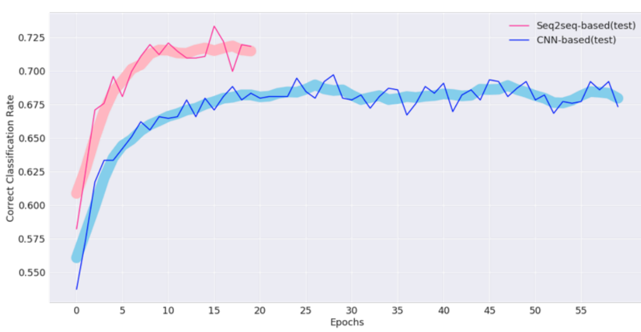

Figure 5. Learning curves of Seq2Seq-based and CNN-based models

D. *Experimental Results*

Table V shows the accuracies of the three DNN models and baseline methods. Overall, the DNN models outperform the baselines, even as the SVMs maintain their high performance. Among baseline methods, the SVM with the RBF kernel achieved the highest accuracy. For the CNN-based models, using word vectors trained using the Wikipedia data slightly enhanced accuracy. For the LSTM-based models, bidirectional processing yielded slightly higher accuracy than single-directional processing.

TABLE V. PREDICTIVE ACCURACIES FOR BASELINES AND DEEP-NEURAL-NETWORK MODELS

Naïve Bayes		SVM(Linear)		SVM(RBF Kernel)	
unigram	*uni+bigram*	*unigram*	*uni+bigram*	*unigram*	*uni+bigram*
0.554	0.598	0.642	0.659	**0.664**	0.659
CNN		**LSTM**		**Seq2Seq**	
with wikipedia	*w.o. wikipedia*	*single-direction*	*bidirection*	*bidirection*	*bidir. w. interm.*
0.686	0.677	0.676	0.678	**0.718**	0.717

There was no significant difference in the accuracies of the CNN model using Wikipedia and the bidirectional LSTM model. Both of these methods outperformed the best of SVMs by 1-2%.

The Seq2Seq model outperformed other methods clearly; the best of SVMs by 5-6% and other DNN models by 3-4%.

The kappa coefficient for the bidirectional LSTM model was 0.63, which is sufficiently high. However, to automatically comprehend and judge the activities of users from only the labels inferred by machines, the kappa coefficient must be improved. By using the Seq2Seq model, which is able to capture the contextual information from the source or the adjacent contribution, the kappa coefficient was improved to 0.723.

Hereafter, we analyze the misclassification of each label individually. The precision and recall for each label are shown in Table VI. Of the ten most frequent labels, the precision of "Greeting" predictions were highest (F1: 0.94) and that of "Agreement" was the second highest (F1: 0.83).

TABLE VI. PRESITION AND RECALL FOR EACH LABEL (RESULT OF BI-DIRECTIONAL LSTM)

Label	Presition	Recall	F1-Value
Agreement	0.85	0.81	0.83
Proposal	0.73	0.74	0.73
Question	0.75	0.8	0.77
Report	0.64	0.62	0.63
Greeting	0.94	0.94	0.94
Reply	0.62	0.46	0.53
Outside Commnets	0.17	0.47	0.25
Confirmation	0.58	0.74	0.65
Gratitude	0.67	0.67	0.67

"Question" was also predicted with high accuracy (F1: 0.77). These results are consistent with our intuition, as both seem to be easy to infer from the contributions themselves, without knowing their context. In contrast, as Table VI shows, the label "Reply" was hard for our model to predict. That performed worst with respect to the recall, tending to be misclassified as an "Agreement", "Proposal" or "Report," as shown in the confusion matrix (Fig. 6). This can be solved if richer context in neighboring contributions is used as input to classifiers in addition to the source contribution.

VI. NEW CODING SCHEME

As indicated in some case that Replay may include a meaning of Agree in the coding scheme based on speech acts used in the current study, the fact that the definition of one label may sometimes overlap the definition of another label has become a factor making it difficult to assign a label always with accuracy and reliability just in artificial intelligence coding but also in manual coding as well. In addition to these technical problems, more importantly, labels based on speech acts which express the linguistic characteristics of the conversation are insufficient for the analysis of the learning process. With this single linguistic scheme, one can not clearly realize whether members of a group engage in activities to solve the task, how members

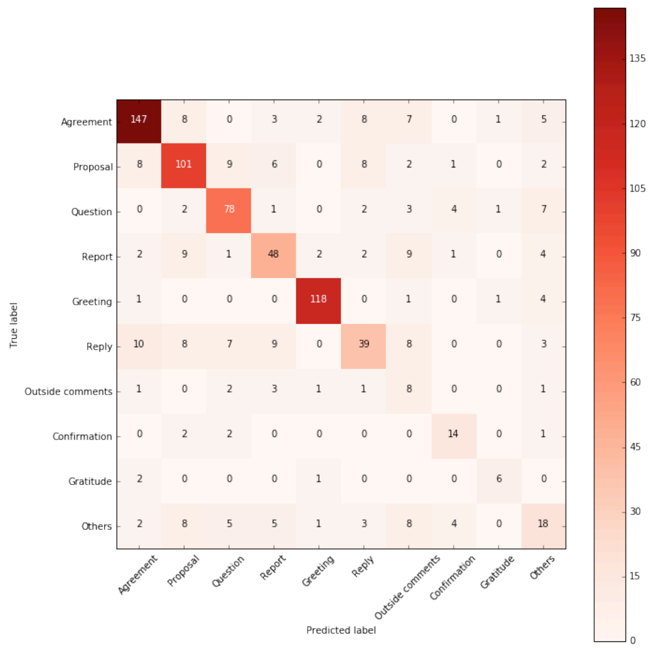

Figure 6. Confusion matrix for the Seq2S2q model.

coordinate each other in terms of task division, time management, etc. during their collaboration, how each member constructs his argument, how members discuss and negotiate each other. From those described above, we propose a new coding scheme so that the automated coding accuracy will improve and that we may understand more accurately and globally collaborative process.

Our new coding scheme is constructed based on the multi-dimensional coding scheme proposed by Weinberger et Fischer who try to analyze whole samples of discourse corpora on multiple process dimensions and "better understand how specific processes of computer-supported collaborative learning contribute to and improve individual acquisition of knowledge" [24]. As shown in Table VII, our scheme consists of five dimensions, while Winberger and Fischer's one has four dimensions without Coordination dimension. We provide labels basically regarding a statement in a chatting as a unit similarly to way we used in the study. In addition, while such values as number of statements are provided as Participation dimension labels, those in other four dimensions are provided by selecting one label from among multiple labels. In other words, since one label is given for each dimension for one statement, a plurality of labels will be assigned to one statement. Therefore, the coding work with this scheme is extremely complicated and takes a lot of time, but the merit of automated coding is even greater. Each dimension is described in detail below.

A. Participation dimension

As shown in Table VIII, Participation dimension is for measuring participation frequency in argumentation. Since this dimension is defined as quantitative data mainly including number of statements, number of letters of statements, time for and interval of statements, there is no need for neither manual nor artificial intelligence coding, requiring a coding just by statistical processing on a database.

Even though Participation dimension labels are capable of analyzing quantitatively different aspects of participation in conversations since they work on specific number of statements or the like, they are incapable of qualitatively analyzing such as whether the contribution has contributed to problem solving.

TABLE VII. NEW CODING SCHEME

Dimension	Description
Participation	Frequency of participation in argumentation
Epistemic	How to be directly involved in problem solving
Argumentation	Ideal assertion in argumentation
Social	How to cope with others' statements
Coordination	How to coordinate to advance discussion smoothly

B. Epistemic dimension

This dimension represents whether each statement is directly related to problem solving as a task and the labels are classified as shown in the table below depending on contents of statements. Labels of this dimension are provided to all statements.

Weinberger and Fischer's scheme has 6 categories to code epistemic activities which consist in applying the theoretical concepts to case information. But, as shown in Table IX, we set only two categories here, because we want to give generality that we can handle as many problem solving types as possible.

TABLE VIII. PARTICIPATION DIMENSION

Category	Description
Number of statements	Number of statements of each member during sessions
Number of letters of a statement	Number of letters during a single speech
Time for statement	Time used for a statement
Interval of statements	Time elapsed since last statement
Statements distribution	Standard deviation of each member within a group

TABLE IX. LABELS IN EPISTEMIC DIMENSION

Label	Description
On Task	Statements directly related to problems
Off Task	Statements without any relationship with problems

"On Task" here indicates such statements which are directly related with assigned problem solving and statements with any of contents described below are regarded as "Off Task."

- Statements asking meaning of problems and how to advance them
- Statements to allocate tasks
- Statements regarding the system

Labels in Epistemic dimension are regarded to be the most basic ones for qualitative analysis since they represent whether they are directly involved in problem solving. For example, it is understood that almost no effort has been made on a problem if there is less "On task" labels.

Besides, Argumentation and Social dimension labels as referred to in the next section and beyond are provided only if Epistemic dimension is "On Task" and those in coordination dimension are provided only if Epistemic dimension is "Off Task."

C. Coordination dimension

Labels of Coordination dimension are provided only if Epistemic dimension labels are "Off Task" and the statements are not directly but indirectly involved in problems. While a list of Coordination dimension labels is shown in Table X, labels are provided not to all of statements of "Off task" but only one label is provided to any statement which falls under the label. For responses to statements to which Coordination dimension labels are provided, those in the same Coordination dimension are provided.

"Task division" here refers to a statement to decide who to work on which task requiring division of tasks for advancing problem solving. "Time management" is a statement to coordinate degree of progress in problem solving, and for example, such statements fall under the definition that "let's check it until 13 o'clock," and "how has it been in progress?" "Meta statement" refers to a statement for clarifying what the problem is when intention and meaning of the problem is not understood. "Technical coordination" refers to questions and opinions about how to use the CSCL System.

TABLE X. LABELS OF COORDINATION DIMENSION

Label	Description
Task division	Allotment of tasks
Time management	Check of temporal and degree of progress
Meta statement	Questions to ask meaning of problems
Technical coordination	How to use the system, etc.

Since Coordination dimension labels are provided to statements for executing problem solving smoothly, it is believed to be possible to predict progress in arguments by analyzing the timing that the labels were provided. In case of less Coordination dimension labels recognized, it is also predicted that smooth relationships have not been built up within the groups.

In a case that a lot of these labels have been provided in many groups, on the other hand, it is assumed that there is some sort of defect in contents of the problems or systems.

In addition, it should be noted that this dimension is not set in Weinberger and Fischer's scheme.

D. Argument dimension

Labels of Argument dimension are provided to all statements when Epistemic labels are "On Task", indicating attributes such as whether each statement includes the speaker's opinion and whether the opinion is based on any ground. Labels of this dimension are provided to just one statement content without considering whether any ground was described in other statement.

A list of Argument dimension labels is shown in Table XI. Here, presence/absence of grounds is determined whether any ground to support the opinion is presented or not but it does not matter whether the presented ground is reliable or not. A qualified claim represents whether it is asserted that presented opinion is applied to all or part of situations to be worked on as a task. "Euphemism" indicates such statements with low confidence rating that presented opinion is just a prediction or shows only possibility. "Non-Argumentative moves" refer to statements without including any opinion and simple questions are also included in this tag.

Labels in Argument dimension are capable of analyzing the logical consistency of statement contents. For example, if a statement is filled just with "Simple Claim" it is assumed as a superficial argument.

In comparison with Weinberger and Fischer's scheme, we introduce a new label "Euphemism". But we do not set for now the categories of macro-level dimension in which single arguments are arranged in a line of argumentation such as arguments, counterarguments, reply, for the reason that it seems difficult that the automatic coding by deep learning methods for this macro dimension works correctly.

TABLE XI. LABELS IN ARGUMENT DIMENSION

Label	Description
Simple Claim	Simple opinion without any ground
Qualified Claim	Opinion based on a limiting condition without any ground
Grounded Claim	Opinion based on grounds
Grounded and Qualified claim	Opinion with limitation based on grounds
Euphemism	Unconfident and ambiguous opinion
Non-argumentative moves	Statement without containing opinion （including questions）

E. Social dimension

Labels in Social dimension are provided when Epistemic code is "On task" but they are provided not to all statements "On task" but to a statement which conforms to Epistemic code. This dimension represents how each statement is related to those of other members within the group. Therefore, it is required to understand not only a statement but also the previous context. A list of this dimension labels is shown in Table XII.

TABLE XII. CODE OF SOCIAL DIMENSION

Label	Description
Externalization	Externalization: No reference to other's opinion
Elicitation	Questionning the learning parner or proviking a reacion from the learning partner
Quick consensus building	Prompt consensus formation
Integration-oriented consensus building	Consensus formation in an integrated manner
Conflict-oriented consensus building	Consensus forming based on a confrontational stance

"Externalization" here refers to a statement without reference to those of others and it is provided mainly to statements as a point of argument origin such as in the beginning of argument on certain topic. "Elicitation" is provided to such statements which require others to extract information such as questions.

From its property as a statement to be made in response to other's opinion, "Consensus building" is classified into the following three labels. "Quick consensus building" is provided to a statement aiming at achieving prompt agreement with other's opinion. In particular, it is provided to a case to agree without delivering any specific opinion. "Integration-oriented consensus building" is provided to statements with an intention to achieve agreement with other's opinion while adding its own opinion. "Conflict-oriented consensus building" is provided to statements which adopt a confrontational stance or request revision against other's opinion.

A sub-dimension called as "Refer" in Social dimension represents which statement is referred to in the statement coded as "Consensus building". Labels in "Refer" dimension are provided without exception only if Social dimension labels belong to "Consensus building."

Since Social dimension labels represent relationship with others, it is possible to estimate how lively discussions were conducted or whose opinion in the group was respected by analyzing Social dimension labels. For example, arguments including a lot of "Quick consensus building" are assumed to be a result obtained just by taking a delivered opinion directly with almost no profound discussion.

F. Each coding and Learning toward artificial intelligence

In the new coding scheme, "Participation" dimension labels are automatically generated from statement logs, whereas other labels require manual coding by a coder in order to build up training data for deep learning and test data. Further, labels to be provided are decided by selecting from any of the dimensions of "Argumentation", "Social" and "Coordination" depending on a result of "Epistemic" labels. Therefore, coder provides "Epistemic" labels based on analysis of "Participation" dimension labels. Subsequently, "Argumentation" and "Social" dimension labels are provided if the "Epistemic labels are "On task." In addition, in a case that "Social" dimension labels belong to "Consensus building", statement number is provided as "Refer" since there exists reference source statement without exception. In a case that "Epistemic" labels are "Off task", those in "Coordination" dimension are provided.

VII. SUMMARY AND FUTURE WORK

This section recapitulates the findings of this study and suggests briefly some future issues.

A. Summary

As the first step to analyze collaborative process of big educational data from the perspective of LA, we tried to automate time-consuming coding task by using deep learning methods.

First, we developed a coding scheme based on the speech acts, coded manually for the remarks, and created training data and test data for deep learning. Next, three DNN models, that is, CNN-based model, LSTM-based model, Seq2Seq-based model were constructed for automatic coding, and their accuracy of automatic coding was verified. In addition, we also compared accuracy with SVMs, which are the baselines of classical machine learning. The result was promising; our approach, particularly, Seq2Seq model outperformed other methods clearly; the best of SVMs by 5-6% and other DNN models by 3-4%. It seems that this model could obtain almost the same predictive accuracy with other coding schemes than ours, for the reason that our coding scheme is sufficiently complex with 16 labels, based not on the surface information, but on the contextual significance of each contribution.

B. Future work

As for the future research directions, we may have two approaches to pursue.

The first approach is about DNN models. To improve prediction accuracy, it may be effective to introduce other network structures such as memory networks [30] instead of DNNs that consist of RNNs and CNNs. Memory networks make a vector from conversation by taking weighted mean of vectors of all sentences. Those weights play a role of attention since they correspond to importance of each sentence. In addition, the context of conversation should be considered. To capture context more precisely, it may be necessary to construct more complex models that take multiple preceding contributions as input vectors.

The second and most important approach concerns coding scheme. Our scheme, based on speech acts, was sufficiently complex, but not global. In order to more accurately and comprehensively grasp various collaborative learning activities such as individual cognitive process, social cognitive process, coordination among members, it will be necessary to construct a coding scheme which is more sensitive to details of interaction and social cognitive process of learning. Therefore, we proposed a new coding scheme with five dimensions, namely the participation dimension, the epistemic dimension, the coordination dimension, the argument dimension, the social dimension. With this new scheme, we are coding all the datasets again to constitute training data and test data for deep learning, in order to verify if this scheme contributes to a more precise understanding of the collaborative process and to improve the accuracy of automatic coding by our DNN models.

ACKNOWLEDGMENT

This work was supported by JSPS KAKENHI Grant Number 26350289 and 16K01134. We also thank to Ryo Yoshinaka for his thoughtful comments.

REFERENCES

[1] C. Shibata, K. Ando, and T. Inaba, "Towards Automatic Coding of Collaborative Learning Data with Deep Learning Technology," The Ninth International Conference on Mobile, Hybrid, and On-line Learning, 2017, pp. 65-71.

[2] G. Stahl, T. Koschmann, and D. Suthers, "Computer-supported collaborative learning," In The Cambridge handbook of the learning science, K. Sawyer, Eds. Cambridge university press, pp. 479-500, 2014.

[3] P. Dillenbourg, P. Baker, A. Blaye, and C. O'Malley, "The evolution of research on collaborative learning," In Learning in humans and machines: Towards an interdisciplinary learning science, P. Reimann and H. Spada, Eds. Oxford: Elservier, pp. 189-211, 1996.

[4] T. Koschmann, "Understanding understanding in action," Journal of Pragmatics, 43, pp. 435-437, 2011.

[5] T. Koschmann, G. Stahl, and A. Zemel, "The video analyst's manifesto (or The implications of Garfinkel's policies for the development of a program of video analysis research within the learning science)," In Video reseach in the learning sciences, R. Goldman, R. Pea, B. Barron and S. Derry, Eds. Routledge, pp. 133-144, 2007.

[6] M. Chi, "Quantifying qualitative analyses of verbal data : A pratical guide ," Journal of the Learning Science, 6(3), pp. 271-315, 1997.

[7] A. Meier, H. Spada, and N. Rummel, "A rating scheme for assesseing the quality of coputer-supported collaboration processes," International Jounal of Computer Suppported Collaborative Learning, 2, pp. 63-86, 2007.

[8] H. Jeong, "Verbal data analysis for understanding interacitons," In The International Handbook of Collaborative Learning, C. Hmelo-Silver, A. M. O'Donnell, C. Chan and C. Chin, Eds. Routledge, pp. 168-183, 2013.

[9] D. Persico, F. Pozzi, and L. Sarti, "Monitoring collaborative activities in computer supported learning," Distance Education, 31(1), pp. 5-22, 2010.

[10] L. Lipponen, M. Rahikainen, J. Lamillo, and K. Hakkarainen, "Patterns of participation and discourse in elementary students'computer-supported collaborative learning," Learning and Instruction, 13, pp. 487-509, 2003.

[11] S. Schrire, "Knowledge building in asynchronous discussion groups: Going beyond quantitative analysis," Computer & Education 46, pp. 49-70, 2006.

[12] 1st Internationa Conference on Learning Analytics and Knowledge. [Online]. Avaiable from: https://tekri.athabascau.ca/analytics/, Nov. 29, 2017.

[13] B. R. Schaun and P. S. Inventado, "Educational data mining and learning analytics," In Learning Analytics, J. A. Larusoon and B. White, Eds. Springer, pp. 61-75, 2014.

[14] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, 521(7553), pp. 436-444, 2015.

[15] Y. Kim, "Convolutional neural networks for sentence classification," arXiv preprint arXiv:1408.5882, 2014.

[16] X. Zhang, J. Zhao, and Y. LeCun. "Character-level convolutional networks for text classification," In Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS2015), pp. 649-657, 2015.

[17] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, 9(8), pp.1735-1780, 1997.

[18] J. Chung, C. Gulcehre, K. Hyun Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," arXiv preprint arXiv:1412.3555, 2014.

[19] Z. Yang et al., "Hierarchical Attention Networks for Document Classification," In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL2016), Human Language Technologies, 2016.

[20] D. Tang, B. Qin, and T. Liu, "Document modeling with gated recurrent neural network for sentiment classification," In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP2016), pp. 1422–1432, 2015.

[21] C. Rosé et al., "Towards an interactive assessment framework for engineering design project based learning," In Proceedings of DETC2007, 2007.

[22] C. Rosé et al., "Analyzing collaborative learning processes automatically: Exploiting the advances of computational linguistics in computer-supported collaborative learning," International Journal of Computer Supported Collaborative Learning, 3(3), pp. 237-271, 2008.

[23] G. Gweon, S. Soojin, J. Lee, S. Finger and C.Rosé, "A framework for assessment of student project groups on-line and off-line," In Analyzing Interactions in CSCL: Methods, Approaches and Issues, S. Putambekar, G.Erkens and C. Hmelo-Silver Eds. Springer, pp. 293-317, 2011.

[24] A. Weinberger and F. Fischer, "A frame work to analyze arugmetative knowledge construciton in computer-supported learning," Computer & Education, 46(1), pp. 71-95, 2006.

[25] B. McLaren, O. Scheuer, M. De Laat, H. Hever and R. De Groot, "Using machine learning techniques to analysze and support mediation of student e-discussions," In Proceedings of artificial intelligence in education, 2007.

[26] T. Inaba and K. Ando, "Development and Evaluation of CSCL System for Large Classrooms Using Question-Posing Script," International Journal on Advances in Software, 7(3&4), pp. 590-600, 2014.

[27] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," arXiv preprint arXiv, pp.1409.0473, 2014.

[28] O. Vinyals and Q. V. Le, " A Neural Conversational Mode," arXiv preprint arXiv:1506.05869, (ICML Deep Learning Workshop 2015), 2015.

[29] T. Kudo, "MeCab: Yet Another Part-of-Speech and Morphological Analyzer". http://mecab.sourceforge.net/, Nov 29, 2017.

[30] S. Sukhbaatar, A. Szlam, J. Weston and R. Fergus, "End-to-end Memory Networks," Proceedings of the 28th International Conference on Neural Information Processing Systems, pp. 2440-2448, 2015.