DOI: 10.33965/celda2019_201911L003

EXAMINING LANGUAGE-AGNOSTIC METHODS OF AUTOMATIC CODING IN THE COMMUNITY OF INQUIRY FRAMEWORK

Yuta Taniguchi[1], Shin'ichi Konomi[2] and Yoshiko Goda[3]
[1]*Faculty of Information Science and Electrical Engineering, Kyushu University, Japan*
[2]*Faculty of Arts and Science, Kyushu University, Japan*
[3]*Research Center for Instructional Systems, Kumamoto University, Japan*

ABSTRACT

This study discusses the automatic coding methods of the Community of Inquiry (CoI) framework for multilingual contexts, in particular. In universities, foreign students cannot be overlooked, and learning systems are also required to work in multilingual situations. However, none of the existing work has addressed the lack of language-agnostic and automatic coding algorithms for the CoI framework, even though the framework is widely used to assess student-generated texts. In this study, we investigate the performance of a data-driven text tokenization algorithm for automatic coding. Using a real-world dataset, we compare the prediction performance of the language-independent tokenizer with a language-dependent tokenizer. Our experiments show the data-driven tokenizer to be comparable to its competitor, and a classification algorithm with this tokenizer could achieve high prediction performance for many CoI indicators. We believe that our experimental results are informative and could provide a baseline for future research.

KEYWORDS

CSCL, CoI Framework, Coding, NLP, Prediction

1. INTRODUCTION

Texts are fundamental elements of an educational context and provide rich information for improving learning and teaching. For instance, reflective learning journals are useful for understanding students' learning, and chat-like messages between students are found in online forums and computer-supported collaborative learning (CSCL) environments. Such chat messages are important as sensors, especially in a CSCL setting, for knowing about the kinds of activities in which students are participating, or how well group work is progressing, etc. To implement an effective CSCL environment, visualization is a useful solution for the enhancement of collaborative learning performance and contributions. A serious problem in CSCL is that of social loafing (Latané et al. 1979), in which some learners are not engaged in collaborative learning activities, because of the non-visualization of their contributions to collaborative work. Therefore, utilizing such textual data is indispensable for tailoring collaborative learning effectively.

One of the important problems associated with such student-generated texts is the automatic coding of texts. Some research exists which has tackled this problem (Zehner et al. 2016), but none of the studies considered the problem in a multilingual setting. Many natural language processing techniques depend on a language's properties; thus, it is not easy to apply an automatic coding algorithm developed for a Western language to an East Asian text dataset. Since education can no longer be separated from the global context, it is crucial that a system work uniformly in any language. Hence, the language-agnostic analysis of educational text data is important and still a challenging task.

The specific focus of this study is the coding of the Community of Inquiry (CoI) framework (Rourke et al. 1999; Garrison, Anderson, and Archer 1999; Anderson et al. 2001). This framework is composed of three elements: social presence, cognitive presence, and teaching presence. The framework has been used in many studies to investigate and evaluate the effect of learning communities and collaborative learning environments (e.g., Yamada 2010; Yamada and Kitamura 2011). Despite their usefulness, automatic coding algorithms of CoI have not been investigated as much. Several studies have indicated that visualization of collaborative learning situations could enhance the awareness of social presence (e.g., Yamada et al. 2016; Yamada et al. 2016). However, categorization accuracy and prediction rate are low, depending on social presence categories. In order to improve the accuracy, automatically-coding algorithm should be improved. For this reason, in this study, we examine methods of coding some CoI components automatically, especially in multilingual situations. Our major concern in this study is tokenization. Tokenization is a type of preprocessing of natural language texts that splits a given sentence into smaller tokens. The most frequently used type of token is the word, since it is relatively easy to identify and considered to be a rich source of meaning. Having said that, words are not always a good unit for characterizing texts, and different units have been proposed (Okanohara and Tsujii 2009; Zhai et al. 2011). Furthermore, tokenization is a fundamental problem in East Asian languages, in which words are not separated by spaces. For such languages, morphological analysis is applied to obtain reasonable sentence splits. However, it is not possible to achieve language-agnostic automatic coding with such analyzers.

Therefore, in this study, we examine the performance of a data-driven tokenization algorithm compared with a Japanese morphological analyzer. Among the many indicators introduced by the CoI framework, we focus on those for social presence and cognitive presence. We formulate the automatic coding problem as a multi-label binary classification problem and then evaluate the prediction performance of some combinations of tokenizers and classifiers with a real-world dataset obtained on our CSCL platform and manually annotated.

2. METHOD

2.1 Classifiers

The task we consider in this study is as follows. There are 13 cognitive presence indicators and 18 social presence indicators, and we want to automatically determine which indicators a given sentence contains, as human annotators do. Since such indicators can be considered as a type of label, our classification task is a general multi-label classification problem. In this study, we adopt the binary relevance approach (Read et al. 2011). With this approach, we regard our task as 31 independent uni-label classification tasks. We train 31 independent uni-label classifiers for each label and then use them to predict each label one by one.

We use two types of classifiers which have shown high performance in text classification tasks. One is random forests (Breiman 2001) classifier, a type of ensemble method based on bagging decision trees. XGBoost (Chen and Guestrin 2016) is the other classifier we use in this study; it is also based on decision trees but uses the gradient boosting technique.

Please note that our experiment aims to compare tokenization methods, not these classifiers, so we do not invest much effort in tuning classifiers, e.g., grid search of hyperparameters. Our experimentation uses existing programming libraries for machine learning, Scikit-learn (Pedregosa et al. 2011), and we mainly use the default parameters of classifiers.

2.2 Tokenization

We compare two tokenization methods: MeCab (Kudo, Yamamoto, and Matsumoto 2004) and SentencePiece (Kudo and Richardson 2018). MeCab is a popular morphological analyzer for tokenizing Japanese sentences. Its model's parameters are pre-trained from a large text corpus and provided as a dictionary. While it can achieve higher tokenization quality for formal texts, it is known to fail with more casual texts. In other words, the tokenization results vary significantly depending on the dictionary, which is why there are some dictionaries for MeCab. Therefore, we expect that it is also hard for MeCab to tokenize chat messages.

SentencePiece is a relatively new tokenization method developed mainly for neural machine translations. It is an unsupervised text tokenizer we can consider as a data-driven approach. It builds a probabilistic language model from an input corpus of texts to tokenize on the fly, and then it segments texts into sub-word pieces. Unlike supervised approaches, this algorithm does not need a pre-tokenized dataset, and thus it is essentially language agnostic. This fact suggests that SentencePiece is more suitable for a text corpus such as ours, in which texts written in different languages are mixed.

Another important point of the SentencePiece algorithm is that it requires the number of tokens to be specified in advance, while this is not required in the case of MeCab. Since this setting is considered to affect the classification results significantly, we try different numbers. We put a limit on the size of vocabularies obtained from both of the tokenizers as follows. In the case of the MeCab tokenizer, we first tokenize all sentences from a corpus and count every token's occurrences. Then, we reject infrequent tokens and choose the most frequent *k* tokens as the vocabulary. However, for the case of SentencePiece, we simply let it choose optimal *k* tokens. We set *k* to 1500, 3500, and 5500.

Prior to tokenization, we apply two normalization processes to all the texts so that we can ignore non-essential variations of texts. One is case conversion, during which we convert all capital letters to their corresponding lower-case letters. The other is Unicode normalization. We employ Normalization Form Compatibility Composition (NFKC) normalization. Through the latter, we can treat different forms of some letters often observed in Japanese texts consistently.

After tokenization, however, we omit the usual preprocessing steps of natural language processing, e.g., stop word removal, in this study, because our purpose is to investigate the possibility of language-independent processing of chat messages, and we want to avoid language-dependent preprocessing as much as possible. Another reason for preserving stop words is because we are *not* classifying chat messages by their topics but by chat method. For example, the phrases "how about" and "what if" are usually deleted as stop words, but they might provide good hints for coding the cognitive presence of "suggestions for consideration" (corresponding to label 24). Therefore, we preserve all of the obtained tokens for better classification results.

2.3 Evaluation

Tokenized chat messages are converted into feature vectors and then passed to a classifier. We use the vector space model as a representation of chat texts and TF-IDF (Salton and Buckley 1988) as a term weighting method. The TF-IDF feature consists of term frequency (TF) and inverse document frequency (IDF). The former works as an indicator of the importance of a word within a document (a chat message, in our case). The latter suppresses topic-irrelevant words, i.e., stop words, and it is important for our settings without stop word removal. Depending on the maximum number of tokens, the resulting feature vectors have different dimensionality.

```
for each label {
    for each split of CV {
        training_dataset, test_dataset = get_dataset_for_this_split()
        for each vocabulary size {
            for each tokenizer {
                tokenized_training_dataset, vocabulary = tokenize_with(tokenizer, training
                tokenized_test_dataset = tokenize_with(tokenizer, est_dataset, vocabulary)
                for each classifier {
                    train(classifier, tokenized_training_dataset)
                    evaluate_with(classifier, tokenized_test_dataset)
                }
            }
        }
    }
}
```

Figure 1. The pseudo code of our evaluation procedure. Please note that we use a vocabulary learned during tokenization of the training dataset for tokenizing the test dataset

Table 1. Labels to be assigned to every chat message. There are two types of labels corresponding to social and cognitive presence indicators. The former is shown in the upper table, while the latter is shown in the lower table. We ignore labels that have no more than 10 positive instances, i.e., five labels are ignored in our experiment

ID	Description	#Positives	Ignored?
1	emotion	293	no
2	humor	26	no
3	selfdisclosure	74	no
4	paralanguage	938	no
5	value	944	no
6	thread	3	yes
7	quoting	1	yes
8	reference	110	no
9	question	652	no
10	appreciation	60	no
11	agreement	384	no
12	disagreement	46	no
13	advice	1	yes
14	vocatives	121	no
15	inclusive	231	no
16	phatics	312	no
17	social sharing	142	no
18	reflection	4	yes

ID	Description	#Positives	Ignored?
19	recognize problem	375	no
20	sense of puzzlement	257	no
21	exploration within the online community	1244	no
22	exploration within a single message	24	no
23	information exchange	527	no
24	suggestions for consideration	308	no
25	leaps to conclusions	216	no
26	integration among group members	76	no
27	integration within a single message	4	yes
28	connecting ideas synthesis	189	no
29	creating solutions	39	no
30	vicarious application to real-world testing solutions	288	no
31	defending solutions	20	no

To evaluate the classification results, cross-validation (Kohavi 1995) is a widely used method that can average the effect of different dataset splits. In this study, we use stratified 10-fold cross-validation. Since each label has a very different number of positive samples, as shown in Table 1, we decided to conduct cross-validation separately for each label. Furthermore, some labels have only a small number of positive instances, and we cannot evaluate classification for them. Therefore, we abandoned evaluations of labels with less than 10 positive instances. As a result, we conduct an evaluation of 26 labels; 14 are social presence indicators, and 12 are cognitive presence indicators. As an evaluation metric, we employ ROC AUC (Fawcett 2006). Figure 1 shows the whole picture of our evaluation procedure. Please note that we use a vocabulary learned during tokenization of the training dataset for tokenizing the test dataset.

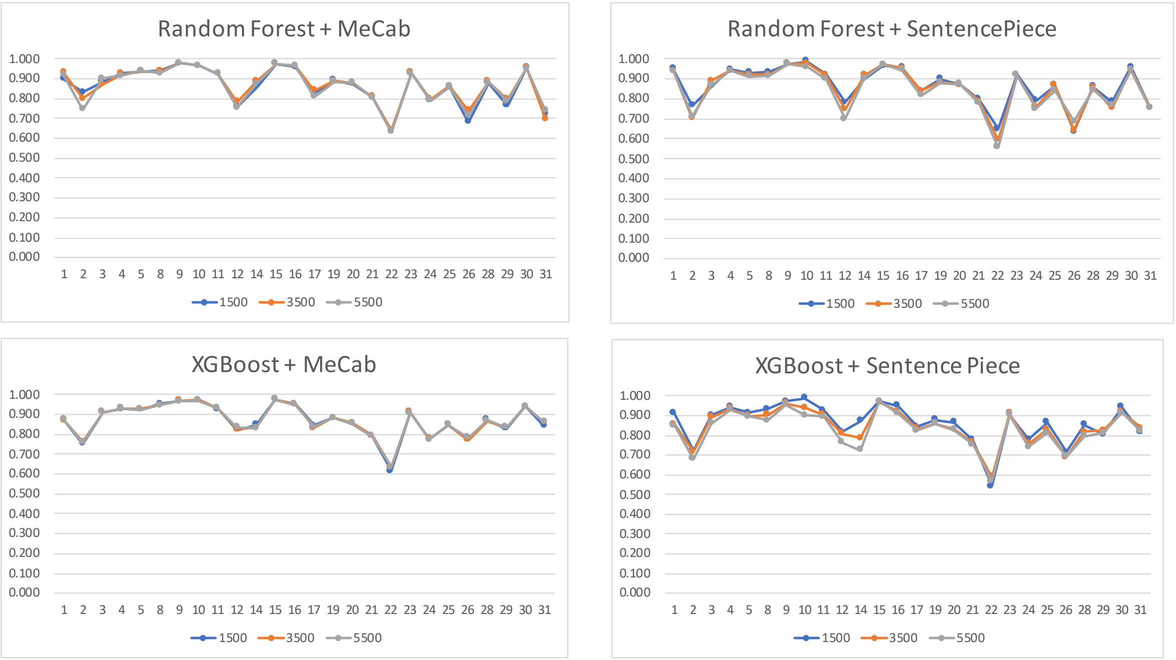
2.4 Labeled Dataset

We use manually labeled chat texts as a training and test dataset for our experiments. Following the coding scheme used in Shea et al. 2010, we give 31 binary labels to every chat message collected on our CSCL platform. The dataset is composed of 3,251 chat messages, written in Japanese and/or English. However, please note that there are messages consisting of words that are neither English nor Japanese. They are written using the alphabet, but the characters are used as phonetic symbols to express the pronunciation of Japanese words. In our group chat system, we use emoticons represented in special notation such as “(smile)” or “(cat)” in the dataset. To prevent them from being tokenized into multiple tokens, we added them to MeCab and SentencePiece tokenizers as pre-defined tokens.

3. RESULTS

Table 2 shows the major outcomes of our experiment. The table shows the area under the receiver operating characteristic curve (AUC ROC) values for every combination of a label (row) and prediction method (column). A prediction method is defined as a combination of three components: a classifier, tokenizer, and the number of tokens. In the table, the darker the background, the better the prediction performance. Again, please note that we do not tune any hyperparameters of the classifiers, and comparison between random forests and XGBoost does not make sense.

First, we compare the results of MeCab and those of SentencePiece. Basically SentencePiece-based predictive models are comparable to MeCab-based ones, and sometimes they have greater performance. Relatively speaking, MeCab-based models have no obvious disadvantages over the competitor, while SentencePiece has some cases of its performance being clearly poorer (see the cases of label 2, label 22, and label 26).


Random Forest + MeCab

Random Forest + SentencePiece

XGBoost + MeCab

XGBoost + Sentence Piece

Figure 2. Visual comparison of the stability of different vocabulary sizes. The most robust combination is XGBoost + MeCab, while the other combinations using SentencePiece are relatively unstable

Second, regarding the effects of vocabulary sizes, we see no large differences in performance, except for certain cases. For example, in the case of the combination of label 12 and SentencePiece and that of label 2 and MeCab, the performance decreases as vocabulary size increases. However, it is difficult to say which vocabulary size we should use for the best performance. In terms of performance stability, Figure 2 provides a visual comparison of the stability of different vocabulary sizes. In the figure, we can easily see the different tendencies: the combinations using SentencePiece are relatively unstable, and, in addition, the combination of XGBoost + MeCab is the most robust.

Table 2. The core outcome from our experiment. In the table, the darker the background, the better the prediction performance. Please note that we do not tune any hyperparameters of these classifiers; comparing them does not make sense

	Random Forests						XGBoost					
	MeCab			SentencePiece			MeCab			SentencePiece		
Label ID	1500	3500	5500	1500	3500	5500	1500	3500	5500	1500	3500	5500
1	0.899	0.931	0.919	0.949	0.940	0.940	0.874	0.871	0.872	0.912	0.857	0.849
2	0.831	0.799	0.747	0.766	0.704	0.712	0.753	0.760	0.759	0.722	0.717	0.681
3	0.885	0.873	0.902	0.860	0.890	0.866	0.910	0.910	0.911	0.901	0.892	0.858
4	0.927	0.925	0.914	0.947	0.940	0.941	0.928	0.931	0.930	0.944	0.936	0.931
5	0.937	0.936	0.939	0.932	0.921	0.913	0.925	0.926	0.924	0.912	0.899	0.894
8	0.940	0.938	0.927	0.933	0.920	0.913	0.951	0.946	0.947	0.934	0.903	0.875
9	0.979	0.978	0.978	0.976	0.974	0.974	0.968	0.968	0.967	0.971	0.960	0.955
10	0.965	0.965	0.966	0.988	0.978	0.959	0.971	0.971	0.970	0.987	0.939	0.902
11	0.925	0.925	0.924	0.922	0.917	0.898	0.927	0.929	0.932	0.926	0.904	0.895
12	0.757	0.783	0.755	0.782	0.748	0.701	0.823	0.824	0.833	0.818	0.807	0.762
14	0.862	0.888	0.875	0.899	0.918	0.902	0.847	0.831	0.830	0.873	0.787	0.724
15	0.975	0.974	0.975	0.967	0.971	0.971	0.975	0.975	0.974	0.971	0.970	0.973
16	0.960	0.964	0.964	0.955	0.954	0.944	0.951	0.953	0.951	0.949	0.921	0.911
17	0.828	0.840	0.808	0.840	0.840	0.817	0.846	0.832	0.838	0.843	0.835	0.824
19	0.892	0.889	0.887	0.899	0.884	0.880	0.884	0.883	0.882	0.877	0.861	0.859
20	0.873	0.875	0.880	0.869	0.870	0.872	0.855	0.856	0.855	0.866	0.831	0.828
21	0.805	0.808	0.807	0.798	0.789	0.780	0.789	0.792	0.789	0.776	0.763	0.754
22	0.638	0.642	0.630	0.648	0.596	0.556	0.614	0.634	0.634	0.542	0.591	0.568
23	0.931	0.932	0.928	0.921	0.918	0.917	0.911	0.910	0.907	0.906	0.910	0.904
24	0.790	0.796	0.789	0.793	0.759	0.752	0.776	0.777	0.775	0.779	0.756	0.741
25	0.858	0.863	0.859	0.867	0.866	0.841	0.847	0.848	0.846	0.866	0.831	0.813
26	0.683	0.740	0.712	0.638	0.643	0.687	0.778	0.770	0.783	0.715	0.689	0.693
28	0.878	0.886	0.882	0.861	0.858	0.846	0.873	0.866	0.870	0.852	0.819	0.795
29	0.764	0.797	0.788	0.786	0.755	0.767	0.827	0.836	0.834	0.804	0.824	0.812
30	0.955	0.956	0.953	0.959	0.947	0.945	0.939	0.940	0.940	0.942	0.918	0.913
31	0.722	0.699	0.739	0.758	0.755	0.755	0.841	0.861	0.860	0.813	0.835	0.820

Finally, comparing labels, we can see relatively clearer differences in predictive performance. Table 2 shows that label 22 is the hardest to predict, followed by labels 2, 12, 26, and 31 is the second hardest. This label corresponds to the indicator "exploration within a single message." Furthermore, when we compare social indicators and cognitive indicators, there are more labels of social indicators in which AUC values are over 0.9 than cognitive indicators. These results suggest that a more advanced analysis of text messages is required, rather than traditional unigram TF-IDF features.

We conclude as follows. 1) We can use SentencePiece as a language-agnostic alternative to MeCab for tokenizing multilingual chat texts; it has comparable performance for providing useful tokens to predictors, despite being an unsupervised algorithm. 2) We need not be as sensitive to vocabulary sizes when using XGBoost with MeCab; however, we must carefully choose one when using SentencePiece as the tokenizer. 3) We found many labels could be predicted fairly well although some labels, especially those of cognitive presence indicators, require a deeper understanding of the texts.

4. CONCLUSION

In this paper, we discussed the multi-label classification problem of group chat messages collected from a collaborative learning support system. Our task was different from usual text categorization problems on two points: 1) our target labels are social presence and cognitive presence indicators, not chat topics, and 2) chat messages consist of both Japanese and English texts requiring language-agnostic treatment. We examined SentencePiece, an unsupervised tokenizer in classification, and our experiments showed that many labels could be predicted with it, although it slightly lacked robustness. We also found that cognitive presence indicators were harder to predict than social presence ones.

Limitations of our research include the facts that the classification method only considers word-level (unigram) features and that no contextual information is considered. Recent developments in natural language processing proposed neural network-based methods which consider contextual information. Such a model could improve the prediction performance of labels that require more contextual consideration. Therefore, our investigation does not present the maximum performance of social and cognitive labeling tasks but merely a baseline. Furthermore, because of the small size of our dataset, five labels with few positive instances were not examined during our experiments. Thus, our conclusions cannot be applied these labels.

ACKNOWLEDGEMENT

This work was supported by JSPS KAKENHI Grant Number JP19H01716.

REFERENCES

Anderson, T., Liam, R., Garrison, D.R. and Archer, W., 2001. Assessing Teaching Presence in a Computer Conferencing Context. *In Journal of Asynchronous Learning Networks*, Vol. 5, No. 2.

Breiman, L., 2001. Random Forests. *In Machine Learning*, Vol. 45, No. 1, pp. 5–32.

Chen, T. and Carlos G., 2016. XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794.

Fawcett, T., 2006. An Introduction to ROC Analysis. *In Pattern Recognition Letters*, Vol. 27, No. 8, pp. 861–874.

Garrison, D.R., Anderson, T. and Archer, W., 1999. Critical Inquiry in a Text-Based Environment: Computer Conferencing in Higher Education. *In The Internet and Higher Education*, Vol. 2, No.2, pp. 87–105.

Kohavi, R., 1995. A Study of Cross-validation and Bootstrap for Accuracy Estimation and Model Selection. *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, Vol. 2, pp. 1137–1143.

Kudo, T. and John R., 2018. SentencePiece: A Simple and Language Independent Subword Tokenizer and Detokenizer for Neural Text Processing. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 66–71.

Kudo, T., Kaoru Y., and Yuji M., 2004. Applying Conditional Random Fields to Japanese Morphological Analysis. *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pp. 230–237.

Latané, B., Williams, K. and Harkins, S., 1979. Many Hands Make Light the Work: The Causes and Consequences of Social Loafing, *In Journal of Personality and Social Psychology*, Vol. 37, No. 6, pp. 822–832.

Okanohara, D. and Jun'ichi T., 2009. Text Categorization with all Substring Features. *Proceedings of the 2009 SIAM International Conference on Data Mining*, pp. 838–846.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. and Vanderplas, J., 2011. Scikit-learn: Machine Learning in Python. *In Journal of Machine Learning Research*, Vol. 12, pp. 2825–2830.

Read, J., Pfahringer, B., Holmes, G. and Frank, E., 2011. Classifier Chains for Multi-label Classification. *In Machine Learning*, Vol. 85, No. 3, p. 333.

Rourke, L., Anderson, T., Garrison, D.R. and Archer, W., 1999. Assessing Social Presence in Asynchronous Text-based Computer Conferencing. *In The Journal of Distance Education*, Vol. 14, No. 2, pp. 50–71.

Salton, G. and Christopher B., 1988. Term-weighting Approaches in Automatic Text Retrieval. *In Information Processing & Management*, Vol. 24, No. 5, pp. 513–523.

Shea, P., Hayes, S., Vickers, J., Gozza-Cohen, M., Uzuner, S., Mehta, R., Valchova, A. and Rangan, P., 2010. A Re-examination of the Community of Inquiry Framework: Social Network and Content Analysis. *In The Internet and Higher Education*, Vol. 13, No. 1, pp. 10–21.

Yamada M. and Kitamura S., 2011. The Role of Social Presence in Interactive Learning with Social Software. *Proceedings of Social Media Tools and Platforms in Learning Environments*, pp. 325–335.

Yamada M., 2010. Development and Evaluation of CSCL Based on Social Presence. *Proceedings of E-Learn: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education*, pp. 2304–2309.

Yamada M., Goda Y., Matsukawa H., Hata K., and Yasunami S., 2016. A Computer-Supported Collaborative Learning Design for Quality Interaction. *In IEEE Multimedia*, Vol. 23, pp. 48–59.

Yamada M., Kaneko K., Goda Y., 2016. Social Presence Visualizer: Development of the Collaboration Facilitation Module on CSCL. *Proceedings of International Conference on Collaboration Technologies*, Vol. 647, pp. 174–189.

Zehner, F., Sälzer, C., and Goldhammer, F., 2016. Automatic Coding of Short Text Responses via Clustering in Educational Assessment. *In Educational and Psychological Measurement*, Vol. 76, No. 2, pp. 280–303.

Zhai, Z., Xu, H., Kang, B. and Jia, P., 2011. Exploiting Effective Features for Chinese Sentiment Classification. *In Expert Systems with Applications*, Vol. 38, No. 8, pp. 9139–9146.