

This is a self-archived version of an original article. This version may differ from the original in pagination and typographic details.

Author(s): Uribe, Pablo; Jiménez, Abelino; Araya, Roberto; Lämsä, Joni; Hämäläinen, Raija; Viiri, Jouni

Title: Automatic Content Analysis of Computer-Supported Collaborative Inquiry-Based Learning Using Deep Networks and Attention Mechanisms

Year: 2020

Version: Accepted version (Final draft)

Copyright: © 2020 The Editor(s) (if applicable) and The Author(s), under exclusive license to S

Rights: In Copyright

Rights url: <http://rightsstatements.org/page/InC/1.0/?language=en>

Please cite the original version:

Uribe, P., Jiménez, A., Araya, R., Lämsä, J., Hämäläinen, R., & Viiri, J. (2020). Automatic Content Analysis of Computer-Supported Collaborative Inquiry-Based Learning Using Deep Networks and Attention Mechanisms. In P. Vittorini, T. Di Mascio, L. Tarantino, M. Temperini, R. Gennari, & F. De la Prieta (Eds.), *Methodologies and Intelligent Systems for Technology Enhanced Learning, 10th International Conference* (pp. 95-105). Springer International Publishing. *Advances in Intelligent Systems and Computing*, 1241. https://doi.org/10.1007/978-3-030-52538-5_11

Automatic Content Analysis of Computer-Supported Collaborative Inquiry-Based Learning Using Deep Networks and Attention Mechanisms

Pablo Uribe¹, Abelino Jiménez¹, Roberto Araya¹, Joni Lämsä², Raija Hämäläinen²,
Jouni Viiri³

¹ Center for Advanced Research in Education, Institute of Education, Universidad de Chile Periodista José Carrasco Tapia 75, Santiago, Chile

² University of Jyväskylä, Department of Education, PO Box 35, FI-40014 Jyväskylä, Finland

³ University of Jyväskylä, Department of Teacher Education PO Box 35, FI-40014 Jyväskylä, Finland

pablo.uribe@student.ecp.fr
roberto.araya.schulz@gmail.com

Abstract. Computer-supported collaborative inquiry-based learning (CSCIL) represents a form of active learning in which students jointly pose questions and investigate them in technology-enhanced settings. Scaffolds can enhance CSCIL processes so that students can complete more challenging problems than they could without scaffolds. Scaffolding CSCIL, however, would optimally adapt to the needs of a specific context, group, and stage of the group's learning process. In CSCIL, the stage of the learning process can be characterized by the inquiry-based learning (IBL) phase (*orientation, conceptualization, investigation, conclusion, and discussion*). In this presentation, we illustrate the potential of automatic content analysis to find the different IBL phases from authentic groups' face-to-face CSCIL processes to advance the adaptive scaffolding. We obtain vector representations from words using a well-known feature engineering technique called Word Embedding. Subsequently, the classification task is done by a neural network that incorporates an attention layer. The results presented in this work show that the proposed best performing model adds interpretability and achieves a 58.92% accuracy, which represents a 6% improvement compared to our previous work, which was based on topic-models.

Keywords: Inquiry Based Learning, Deep Neural Networks, Natural Language Processing.

1 Introduction

Scholars widely agree that lecture-based teaching should be complemented with more active learning methods to support the development of skills and knowledge that stu-

dents graduating from science, technology, engineering, and mathematics (STEM) domains need [1,2]. In this respect, the potential of computer-supported collaborative inquiry-based learning (CSCIL) has been known for a long time [3], and it still is a popular pedagogical approach to enhance skills and knowledge beneficial for future STEM professionals [4]. In short, CSCIL is a technologically facilitated and mediated process in which a group of students follows the practices of scientists to acquire scientific knowledge, learn scientific content, and better understand the nature of science [5]. CSCIL emphasizes the student's active role in the learning process; students are encouraged to explore the material, ask questions, and share ideas with each other so that technological advancements can increase the success of learning even more [6].

CSCIL is not an unambiguous pedagogical method or model, and there is no unified theory of CSCIL. Pedaste et al. [7], however, have made a synthesis of the various inquiry-based learning (IBL) models. They provided a framework in which the essential aspects of IBL are captured with the help of five phases—orientation, conceptualization, investigation, conclusion, and discussion. In the orientation phase, students should identify the main concepts and variables of the problem and become familiar with the needed technological resources. In the conceptualization phase, students should determine the dependent and independent variables as well as propose research questions or hypotheses that they start to investigate. In the investigation phase, students should plan their data collection procedure, implement the procedure, and analyze and interpret the data. In the conclusion phase, students should offer and evaluate solutions to their questions or hypotheses. In the discussion phase, students should elaborate on their findings and conclusions as well as reflect their CSCIL. Even though collaboration and technological resources themselves can assist students in IBL, research has shown that other scaffolds are also needed to achieve the benefits of CSCIL [8]. It is also known that the needs for scaffolds are different in the different IBL phases [5]. Thus, before designing and implementing the scaffolds, there is a need to study CSCIL with particular focus on the IBL phases. To study CSCIL, researchers can conduct content analysis [9,10], for example, so that they code the transcribed students' conversations to the different IBL phases (orientation, conceptualization, investigation, conclusion, and discussion) [7].

Currently, researchers conduct content analysis procedures mostly manually. The human-driven content analysis of large data sets, however, is time-consuming. Moreover, the validity of inferences from the data depends on the consistency of the coding procedures [11], which is why the inter-coder and intra-coder reliability are subject to intense methodological research efforts over long years [12]. The development of an automatic content analyser could have significant implications concerning the scaffolding CSCIL. First, automation allows large-scale analyses. Second, it might enable the real-time monitoring of several groups when they engage in CSCIL. The real-time information about groups' ongoing IBL phase could be useful so that technological learning environments or teachers could adapt scaffolds based on each group's needs. The present work introduces an automatic content analysis method for utterance classification so that the IBL phase can be automatically captured from CSCIL processes taking place in face-to-face interaction in an authentic higher education setting. Our method shows the potential of computer-driven analysis to address the current challenges of

manual content analysis, namely insufficiency of the time resources and issues concerning reliability. We address the following research question: How similar are the results of the manual and proposed automatic content analysis?

2 Related Work

The present work focuses on the automatization of the IBL phase coding necessary for all further analysis. Therefore, this work contributes by automatizing a time-consuming process of researchers' work, so researchers can focus on interpreting results and designing optimal scaffolds. A previous work done by our team presented in [13] had the same objectives and was used as a guide-line for this research. To improve the performance, the methodology presented here focuses on two points:

Feature Engineering. The previous work was based on a Latent Dirichlet Allocation (LDA) topic model [14]. Topic models are statistical models that are used to find groups of words, called topics, that usually appear together in large document collections [13]. This model was trained with scientific literature (physics textbooks) to generate features from the utterances, representing them as a distribution of a fixed list of 60 learned topics. However, the LDA model training process did not include natural dialogic language sources that are present in common social interactions (such as groups' conversations), which are however difficult to obtain. Nevertheless, even with these limitations the results of the previous work were promising. Alternatively, in this work we used a Word Embedding model. This procedure consists in the assignment of high dimensional vectors to words in a way that preserves the syntactic and semantic relationships between them, and is one of the most fundamental techniques in natural language processing [15]. When trained on large enough corpora, this model admits vector representations for a big number of words, even for typical of dialogic language. In this work, we evaluate a word embedding model already trained on a mass scale corpus (provided by the *TurkuNLP* project [16]), to obtain a numerical representation of utterances as sequences of vectors.

Classification Algorithm. The preceding study used Support Vector Machines (SVMs) [17] trained with the hand-labelled transcriptions from the groups' conversations to classify each utterance. Instead, we used a deep neural network with an embedding and an attention layer, which are widely applied in the Natural Language Processing tasks [18]. When needed, the incorporation of an embedding layer makes possible the adjustment of the word embedding vectors during the training process. On the other hand, attention layers are a standard part of the deep learning toolkit, contributing to impressive results in various tasks. In fact, a standard neural network consists of a series of non-linear transformation layers, where each layer produces a fixed-dimensional hidden representation. For tasks with large input spaces, this paradigm makes it hard to control the interaction between components. However, an attention network maintains a set of hidden representations that scale with the size of the source by performing a soft-selection over these representations, as explained in [19]. In this work we implement two attention mechanisms. The first is called *simple attention*, that soft-selects important words from each utterance in a general manner for all categories. The

second operates in a category-specific way, so each IBL phase performs the soft-selection according to their own nature. This mechanism is called *differentiated attention*. Further details will be discussed in the next section.

3 Methodology

In this study, we analysed 55 students in an introductory university physics course on thermodynamics. The participants were divided into eleven groups of five students and each group worked with a shared laptop computer. The students were asked to collaboratively solve thermodynamics problems in a technology-enhanced learning environment while their conversations were screen-captured and audio recorded. In this section, we present the procedures when implementing our model for automatic content analysis.¹

3.1 Data Set

Our data set was built by manually transcribing each group’s talk while they solved an inquiry problem. The transcriptions are in Finnish. Each group said, on average, 180 utterances, summing up to 1980 for the whole data set. These utterances include, on average, 11 words. The utterances were manually labelled by using theory-driven content analysis [20], i.e. each utterance was coded to one of the IBL phase presented by [7]. One of the researchers coded all the utterances while another researcher outside of this study independently coded 20% of the utterances. The inter-rater agreement was 67.7%, after which the disagreements were discussed and resolved.

3.2 Data Pre-processing

Text data is pre-processed to transform it into a simpler form so algorithms can perform better. First, raw digits are converted to words (example: ‘92’ is turned into ‘ninety-two’). Second, punctuation marks are removed, except from questions marks that are considered as new words. Later, a tokenization is done considering only the top 2000 most frequent words (out of 3500 different words found in the transcriptions). Finally, utterances are transformed to the same fixed length of 20 words. As a consequence, a total of 254 utterances are truncated, and 1700 are padded with the token 0.

3.3 Feature Engineering

In this work, the main input for the model is the current utterance represented as a sequence of tokens. This sequence will be later transformed into a sequence of word-vectors by the embedding layer explained later. Additionally, we considered as input for our model the previous and the next utterances’ token-sequence representations, as

¹ Further details are available at <https://github.com/pabloveazul/CIBL>.

well as the number of words of the current utterance and its relative position in the respective group work session.

3.4 The Neural Network Classifier

In this work we have replaced the precedent SVM with a neural network composed of different layers. Distinct configurations of these layers give rise to numerous models with different architectures that are evaluated later. The main layers are explained below²:

Embedding Layer. Word embedding vectors are obtained from the *TurkuNLP* project, where an already trained word embedding model is available for public use. These vector representations are obtained using a word2vec model [21] trained on the *Finnish Internet Parsebank* (FIB), a mass-scale corpus with automatic syntactic analysis that currently includes about 3.7 billion tokens [16]. This layer turns tokens into word embedding vectors. We then represent utterances as ordered sequences of word-vectors. Mathematically, each word token m in the vocabulary is associated with an d -dimensional embedding $Emb(m) \in R^d$. If m is represented by its one-hot vector \mathbf{m} , $Emb(m)$ corresponds to the column \mathbf{Em} of an embedding matrix $\mathbf{E} \in R^d \otimes R^{|V|}$. Here $|V|$ is the size of the vocabulary V ($|V|=2000$) and d is the dimension of the word embedding ($d=200$). Then, the embedding layer output of a sequence of 60 words $\mathbf{u} = [\mathbf{m}_1, \dots, \mathbf{m}_{60}]$, corresponding to the previous, current and next utterance (each one of 20 words) is given by: $\mathbf{e} = [\mathbf{Em}_1, \dots, \mathbf{Em}_{60}]$. The matrix weights $E_{i,j}$ are initialized with the weights given by the TurkuNLP project, and can be adjusted through back-propagation during the training process (if they are set to trainable) or remain constant (if they are set to static). The embedding of the token 0 is the null vector.

Attention Layer. To enhance relevant words for the classification task, a simple attention mechanism is built. A Single Layer Perceptron (SLP) with a single output is applied to every temporal slice of the encoded sequence (i.e. to each embedding of \mathbf{e}). The full output of this layer is called the attention weight vector: $\mathbf{w} = [w_1, \dots, w_{60}]$. Mathematically, the attention weight of each encoded word \mathbf{e}_t is given by:

$$w_t = SLP(\mathbf{e}_t) = \mathbf{a}^T \mathbf{e}_t + b \quad (1)$$

Here, \mathbf{a} is a vector with the same dimension as \mathbf{e}_t and $b \in R$. These are the parameters of the SLP which are learnt through backpropagation. Later, a softmax layer is applied time-wise (i.e. word-wise) to obtain probabilities proportional to the exponentials of the weight numbers. The output is then interpreted as an attention-probability vector $att(\mathbf{e}) = [att(\mathbf{e})_1, \dots, att(\mathbf{e})_{60}]$:

$$att(\mathbf{e})_t = softmax(\mathbf{w})_t = e^{(\mathbf{a}^T \mathbf{e}_t + b)} / (\sum_{i=1}^{60} e^{(\mathbf{a}^T \mathbf{e}_i + b)}) \quad (2)$$

This attention-probability vector is then multiplied element-wise with the vector $\mathbf{e} = [\mathbf{e}_1, \dots, \mathbf{e}_{60}]$ of encoded words to finally obtain a weighted sequence

² All vectors and matrix written in bold, scalars in light.

$[att(\mathbf{e})_1 \mathbf{e}_1, \dots, att(\mathbf{e})_{60} \mathbf{e}_{60}]$. Each attention probability can be interpreted as the importance of each word in the utterance. This attention mechanism is called simple attention. However, one may think that the importance of words may vary depending on the final classification. For example, numbers within an utterance should be the important features for the investigation phase, whether concept words should be respectively for the conceptualization phase. This leads to another possible architecture for the attention layer, where each category is connected with one independent attention mechanism. This means each category c is associated with one SLP (noted SLP_c) with a single output, and no parameters are shared between these SLPs. Mathematically this is:

$$w_t^c = SLP_c(\mathbf{e}_t) = (\mathbf{a}^c)^T \mathbf{e}_t + b^c \quad (3)$$

Where (\mathbf{a}^c) and b^c correspond to the parameters of the SLP associated to the category c . Similarly, the attention-probability of a given category c and a given encoded word \mathbf{e}_t is then computed as:

$$att_c(\mathbf{e})_t = softmax(\mathbf{w}^c)_t = e^{(\mathbf{a}^c)^T \mathbf{e}_t + b^c} / (\sum_{i=1}^{60} e^{(\mathbf{a}^c)^T \mathbf{e}_i + b^c}) \quad (4)$$

The output of this layer are the five weighted sequences: $[att_c(\mathbf{e})_1 \mathbf{e}_1, \dots, att_c(\mathbf{e})_{60} \mathbf{e}_{60}]$. This attention mechanism is named *differentiated attention*.

Sum Layer. A simple sum is applied timewise over all the vectors from the previous layer to obtain a context vector $\mathbf{s} = \sum_{t=1}^{60} \mathbf{u}_t$, where $\mathbf{u} = [\mathbf{u}_1, \dots, \mathbf{u}_{60}]$ is the previous output sequence. When the differentiated attention mechanism is present, this sum is done independently on each weighted sequence of each category c : $\mathbf{s}_c = \sum_{t=1}^{60} att_c(\mathbf{e})_t \mathbf{e}_t$.

Multi-Layer Perceptron. After the input corresponding to the utterances is processed by the previous layers, the final sum s is concatenated with the inputs corresponding to the number of words of the utterance (n) and its relative position in the CSCIL-process (r) into a single vector $[s, n, r]$. This vector is fed into an MLP with one hidden layer, and an output (\mathbf{y}) of dimension 5 (one for each IBL phase) with softmax activation function to obtain a probability distribution over the different phases as a final prediction $\hat{\mathbf{y}}$. This prediction is interpreted as the probability of the utterance to belong to each one of the phases.

$$\mathbf{y} = MLP([s, n, r]) \quad (7)$$

$$\hat{\mathbf{y}} = softmax(\mathbf{y}) \quad (8)$$

In the case of the differentiated attention mechanism, each sum s_c is concatenated with n and r into a single vector $[s_c, n, r]$ and fed into a category-specific MLP. Each MLP (noted MLP_c) is independent and has a hidden layer and a single output y_c . A softmax layer is applied then to the concatenation vector of these outputs to obtain the final prediction $\hat{\mathbf{y}}$. This is:

$$y_c = MLP_c([s_c, n, r]) \quad (9)$$

$$\hat{y} = \text{softmax}([y_1, y_2, y_3, y_4, y_5]) \quad (10)$$

3.5 Model Evaluation

To evaluate each model, we independently split the training and testing sets with all possible combinations of nine and two group transcriptions respectively. Models are trained through backpropagation using an ADAM optimizer [22] and cross-entropy loss function. Then, the average accuracy over all the splitting combinations is considered as the performance of each model.

4 Results

Several models with different layers are evaluated. A table with the description and the results of the model evaluation process are shown in the Table 1 (the best results in bold):

Table 1. Model description and evaluation results. The confidence interval is calculated with the 95% confidence level. (**Pre-Trained Trainable**: initiated with TurkuNLP weights and trained during training process, **Pre-Trained Static**: initiated with TurkuNLP weights and not trainable, **No**: no addition of the corresponding layer. Model names **MLP**: Multilayer Perceptron, **SE**: Static Embedding Layer, **TE**: Trainable Embedding Layer, **SA**: Simple Attention, **DA**: Differentiated Attention)

Model Name	Embedding Layer	Attention Layer	Mean Accuracy on Test Set	Mean Accuracy on Training Set
MLP+SE	Pre-Trained Static	No	54.43 ± 0.90	76.46 ± 2.02
MLP+SESA	Pre-Trained Static	Simple	57.38 ± 1.02	61.87 ± 0.99
MLP+SEDA	Pre-Trained Static	Differentiated	58.92 ± 0.96	64.57 ± 1.34
MLP+TE	Pre-Trained Trainable	No	54.33 ± 1.07	85.87 ± 2.78
MLP+TESA	Pre-Trained-Trainable	Simple	56.03 ± 0.98	69.50 ± 1.98

A comparison between the manual and automatic classification of the best performing model (MLP+SEDA) is presented through a confusion matrix. Each component $C_{i,j}$ in the matrix is the average percentage of utterances in the test set that were manually coded to IBL phase i and automatically coded to IBL phase j :

Table 2. Mean Confusion Matrix of the best performing model (Model 3). For each real phase, the highest percentage between the phase predictions is in bold.

Real Phase	Predicted Orientation	Predicted Conceptualization	Predicted Investigation	Predicted Conclusion	Predicted Discussion
Orientation	17	0.87	1.7	0.12	5.6
Conceptualization	2.4	2.1	1.4	0.07	5

Investigation	2.6	0.48	16	0.02	5.9
Conclusion	0.04	0.25	0.52	0.2	2.7
Discussion	6	0.78	4.2	0.4	24

A comparison table between the previous LDA model [6], the MLP+SEDA and a human coder regarding the precision for each phase is shown below:

Table 3. Precision of different IBL phase utterance classifiers. For each phase, the highest precision between the automatic models is in bold.

Classifier	Orientation Precision (%)	Conceptualization Precision (%)	Investigation Precision (%)	Conclusion Precision (%)	Discussion Precision (%)
LDA	50	49	68	49	51
MLP+SEDA	60	47	67	24	55
Human Coder	71	55	76	45	64

Attention probabilities of the differentiated attention mechanism incorporated to this model let us understand what type of words are relevant for each IBL phase. In Figures. 1. and 2. we graphically represent the attention probabilities for each word in the previous, current and next utterance respectively. In the x-axis each word of each utterance is written according to the previous order. Blank spaces correspond to the null word vectors added by the padding process. The y-axis represents the attention probability. Each coloured line corresponds to a different category (IBL phase):

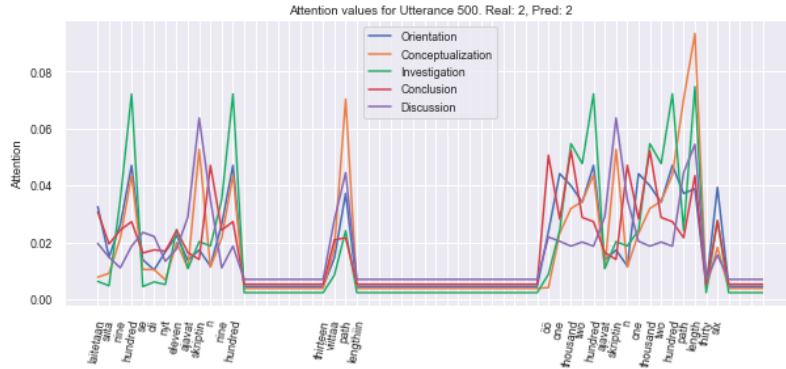


Fig. 1. MLP+SEDA correctly classifying the utterance 500 into the investigation phase. Attention probabilities corresponding to the investigation phase are high for numbers ('nine', 'two', 'hundred', 'thousand').

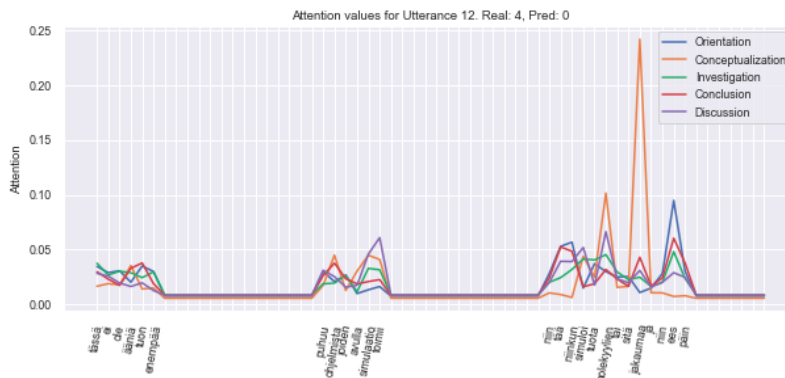


Fig. 2. MLP+SEDA misclassifying the utterance 12 into the orientation phase (real: discussion). Attention probabilities of the conceptualization phase are high for concept words ‘molekyyliin’ (molecules) and ‘jakaumaa’ (distribution).

Figures 1 and 2 show that the attention probability corresponding to null vectors is close to 0 for every IBL phase. Also, the attention probabilities effectively vary between the different IBL phases, so that peak values are found for a different type of words in each case.

5 Discussion and Conclusion

This work was our second attempt to automatize the content analysis in an authentic CSCIL context. We compared the results with the ones discussed in [13], which we have exceeded using the new models by 6%, achieving a 58.9% average accuracy (the previous model has a 52.9% average accuracy). Nevertheless, there is still a challenge to improve the automatic content analysis models to attain human-level performance (67% accuracy). Additionally, most of the issues found for the previous results remain present for the ones obtained in this work: precision is still notably higher for the investigation phase. On the other hand, despite improving the precision for the orientation and discussion phase, the precision of all the other IBL phases decreased. To improve the precision of the conceptualization and conclusion phases (see Table 2), we will gather more data as we now constrained ourselves to a thermodynamics problem.

Regarding the methodology, we replaced the previous SVM classifier with a deep network that can model more complex functions. Also, it incorporates an embedding layer that associates high-dimensional features to words, which can be trained through backpropagation. Therefore, the previous feature engineering manual process based on an LDA model is replaced by this procedure. Additionally, the pre-trained embedding model given by the Turku University NLP project incorporates face-to-face Finnish vocabulary that was only partially contained previously in the physics textbooks. For instance, adding an attention layer helps not only to improve performance but also to obtain more interpretable results by analysing the attention weights. For each IBL

phase, words with high weights may be interpreted as the key elements for the classification task within each utterance.

The automatized identification of the IBL phase from students' face-to-face conversation could be used for adaptive scaffolding purpose. Even though the idea of the adaptive scaffolding is not a new one [23], the work is still in progress [24]. Our results may provide input for this development of systems that allow technological learning environments or teachers monitor in real-time (through a dashboard in their smartphones or notebooks) the IBL phase of several groups' CSCIL processes. These systems may include other applications, such as giving quick feedback to teachers regarding their speech when they provide support to students, as presented in [25].

6 Acknowledgements.

Support from ANID/ PIA/ Basal Funds for Centres of Excellence FB0003 is gratefully acknowledged. We would like to express our gratitude for the financial aid provided by the Academy of Finland [grant numbers 292466 and 318095, the Multidisciplinary Research on Learning and Teaching profiles I and II of JYU].

References

1. Arthurs, L. A., & Kreager, B. Z. (2017). An integrative review of in-class activities that enable active learning in college science classroom settings. *International Journal of Science Education*, 39(15), 2073–2091.
2. Freeman, S., Eddy, S. L., McDonoug, M., Smith, M. K., Okoroafor, N., Jordt, H., & Wenderoth, M. P. (2014). Active learning increases student performance in science, engineering, and mathematics. *Proceedings of the National Academy of Sciences*, 111(23), 8410–8415.
3. Lipponen, L., & Hakkarainen, K. (1997). *Developing culture of inquiry in computer-supported collaborative learning* (pp. 171–175). Toronto, Ontario, Canada: International Society of the Learning Sciences.
4. Jeong, H., Hmelo-Silver, C. E., & Jo, K. (2019). *Ten years of computer-supported collaborative learning: A meta-analysis of CSCL in STEM education during 2005–2014*. *Educational Research Review*, 28, 100284.
5. Bell, T., Urhahne, D., Schanze, S., & Ploetzner, R. Collaborative inquiry learning: Models, tools, and challenges. *International journal of science education* 32(3), 349-377 (2010)
6. Pedaste, M., Leijen, Ä., Saks, K., De Jong, T., & Gillet, D. (2017, January). How to link pedagogy, technology and STEM learning? In *the proceedings of the 25th International Conference on Computers in Education* (pp. 579-586).
7. Pedaste, M., Mäeots, M., Siiman, L. A., De Jong, T., Van Riesen, S. A., Kamp, E. T., ... & Tsourlidaki, E. Phases of inquiry-based learning: Definitions and the inquiry cycle. *Educational research review* 14, 47-61 (2015).
8. Alfieri, L., Brooks, P. J., Aldrich, N. J., & Tenenbaum, H. R. Does discovery-based instruction enhance learning? *Journal of Educational Psychology* 103(1), 1-18 (2011)
9. Lämsä, J., Hämäläinen, R., Koskinen, P., & Viiri, J. Visualising the temporal aspects of collaborative inquiry-based learning processes in technology-enhanced physics learning. *International Journal of Science Education* 40(14), 1697-1717 (2018)

10. Lämsä, J., Hämäläinen, R., Koskinen, P., Viiri, J., & Mannonen, J. The potential of temporal analysis: Combining log data and lag sequential analysis to investigate temporal differences between scaffolded and non-scaffolded group inquiry-based learning processes. *Computers & Education* 143, 103674 (2020)
11. Weber, R. P.: *Basic content analysis* (No. 49). Sage (1990).
12. Krippendorff, K.: *Content analysis: An introduction to its methodology*. Sage publications (2018).
13. Espinoza, C. Joni Lämsä, Roberto Araya, Raija Hämäläinen, Abelino Jiménez, Raúl Gormaz, and Jouni Viiri: Automatic content analysis in collaborative inquiry based learning, ESERA, 2019.
14. Blei, D. M., Ng, A. Y., & Jordan, M. I. Latent dirichlet allocation. *Journal of machine Learning research* 3(Jan), 993-1022 (2003).
15. Tshitoyan, V., Dagdelen, J., Weston, L., Dunn, A., Rong, Z., Kononova, O., ... & Jain, A. Unsupervised word embeddings capture latent knowledge from materials science literature. *Nature* 571(7763), 95-98 (2019).
16. Turku NLP page, Finnish NLP https://turkunlp.org/finnish_nlp.html#treebank, last accessed 2020/01/29.
17. Burges, C. J. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery* 2(2), 121-167 (1998).
18. Hu D. An Introductory Survey on Attention Mechanisms in NLP Problems. In: Bi Y., Bhatia R., Kapoor S. (eds) Intelligent Systems and Applications IntelliSys 2019. Advances in Intelligent Systems and Computing, vol 1038, pp 432-448. Springer, Cham (2020).
19. Kim, Y., Denton, C., Hoang, L., & Rush, A. M. Structured attention networks. arXiv preprint arXiv:1702.00887, (2017).
20. Neuendorf, K. A.: *The content analysis guidebook*. Sage (2016).
21. Mikolov, T., Chen, K., Corrado, G., & Dean, J. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, (2013).
22. Kingma, D. P., & Ba, J. Adam. A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, (2014).
23. Kollar, I., Fischer, F., & Slotta, J. D. (2007). Internal and external scripts in computersupported collaborative inquiry learning. *Learning and Instruction*, 17(6), 708–721.
24. de Jong, T. (2019). Moving towards engaged learning in STEM domains; there is no simple answer, but clearly a road ahead. *Journal of Computer Assisted Learning*, 35(2), 153–167.
25. Caballero D. et al. ASR in Classroom Today: Automatic Visualization of Conceptual Network in Science Classrooms. In: Lavoué É., Drachsler H., Verbert K., Broisin J., Pérez-Sanagustín M. (eds) Data Driven Approaches in Digital Education EC-TEL 2017, LNCS, vol 10474, pp 541-544. Springer, Cham (2017).