Automatic analysis of cognitive presence in online discussions: An approach using deep learning and explainable artificial intelligence

Yuanyuan Hu a,*, Rafael Ferreira Mello b, c, Dragan Gašević d, e, f

Abstract
This paper proposes the adoption of a deep learning method to automate the categorisation of online discussion messages according to the phases of cognitive presence, a fundamental construct from the widely used Community of Inquiry (CoI) framework of online learning. We investigated not only the performance of a deep learning classifier but also its generalisability and interpretability, using explainable artificial intelligence algorithms. In the study, we compared a Convolution Neural Network (CNN) model with the previous approaches reported on the literature based on random forest classifiers and linguistics features of psychological processes and cohesion. The CNN classifier trained and tested on the individual data set reached results up to Cohen’s $\kappa$ of 0.528, demonstrating a similar performance to those of the random forest classifiers. Also, the generalisability outcomes of the CNN classifiers across two disciplinary courses were similar to the results of the random forest approach. Finally, the visualisations of explainable artificial intelligence provide novel insights into identifying the phases of cognitive presence by word-level relevant indicators, as a complement to the feature importance analysis from the random forest. Thus, we envisage combining the deep learning method and the conventional machine learning algorithms (e.g. random forest) as complementary approaches to classify the phases of cognitive presence.

1. Introduction
Asynchronous online discussions play a key role in aiding social interaction within educational environments, especially in higher education (Anderson & Dron, 2010). They promote course participation, collaboration, and even reduce drop-out rates (Rivard, 2013). Although the online discussions have several benefits, it is crucial to provide tools and methods to assist the instructors to monitor contributions and promote understanding of the process of knowledge (co-)construction in a group of students (Garrison, Cleveland-Innes, & Fung, 2010).

In this context, the Community of Inquiry (CoI) framework (Garrison et al., 1999) is a widely validated model that provides different dimensions to analyse the dynamic and social nature of modern online learning. CoI defines three presences that explain students and instructors’ social interactions in online learning environments. Within them, cognitive presence is a central construct to seize the development of the critical, in-depth thinking and argumentation skills of the students (Garrison et al., 2001). The CoI framework also defines coding schemes for the categorisation of discussion messages of students according to each presence. For instance, cognitive presence is operationalized using a four-phase inquiry process (Garrison et al., 2001): 1) Triggering event, where a problem or dilemma is identified and conceptualized, 2) Exploration, in which students’ explore and brainstorm potential solutions to the issue at hand, 3) Integration, during which students (co-) construct new knowledge by synthesizing the existing information, and 4) Resolution, in which students evaluate the newly-created knowledge through hypothesis testing or vicarious application to the problem/dilemma that triggered the learning cycle. Although commonly used in research for content analysis, the manual coding process is labour-intensive, time-consuming, and not practical to be performed on...
ongoing discussions to assist instructors in their teaching (Kovanović, Gašević, & Hatala, 2014).

Several automated and semi-automated methods have been proposed in the recent literature (Barbosa et al., 2020; Ferreira et al., 2018; Kovanović et al., 2014b, 2016; Neto et al., 2018; Rolim et al., 2019; Waters et al., 2015). While early approaches applied a combination of word and phrase counts features with black-box machine learning algorithms (i.e., Neural Networks and Support Vector Machine) (Kovanović, Joksimović, et al., 2014; Mcklin, 2004), recent studies proposed the adoption of features based on psychological processes, writing cohesiveness, and discussion structure compound with decision tree algorithms (Barbosa et al., 2020; Kovanović, Joksimović, et al., 2014; Neto et al., 2018). The main advantages of the second approach are the description of important indicators of different cognitive presence phases and the improvement in the generalisability of the created classifiers (Kovanović et al., 2016).

Advances in natural language processing have led to the development of more accurate text classification techniques based on deep neural networks (Lai et al., 2015). The main idea of these methods is that the classifier tends to reach better results, even for different contexts, as the database increases. Moreover, explainable artificial intelligence is a recent method for unpacking the results of deep neural networks classifiers (Miller, 2019; Samek & Müller, 2019). The combination of these approaches could provide alternative analysis not explored before in terms of the analysis of CoI constrains.

Therefore, this paper proposes a new method for automatic categorisation of online discussion messages using deep machine learning and explainable artificial intelligence. Our study analyses the efficiency and the generalisability of this approach in the context of cognitive presence. The results showed that this approach reached results comparable to the previous approaches reported on the literature (Barbosa et al., 2021; Kovanović et al., 2016; Neto et al., 2018). Moreover, the explainable artificial intelligence visualisations shine new insights into the feature importance for this problem. The results and their implications are further discussed in this paper.

2. Background

2.1. The Community of Inquiry and cognitive presence

The Community of Inquiry (CoI) framework proposed by Garrison et al. (1999) is a social-constructivist model used to analyse the interactive educational experience in online learning environments. It describes educational engagement that occurs in a learning community, in which “a group of individuals who collaboratively engage in purposeful critical discourse and reflection to construct personal meaning and confirm mutual understanding” (Garrison & Anderson, 2011, p. 2). The CoI framework has been widely used and validated for multidisciplinary courses to understand the participants’ engagement in online discussion forums (Garrison & Anderson, 2011; Garrison, Anderson, & Archer, 2010; Hu et al., 2020; Liu & Yang, 2014). This model defines three dimensions, called presences, to mould the learning experience: (i) Cognitive presence explains the critical reflection of the knowledge (re)construction, and problem-solving processes in the learning community (Garrison et al., 2001); (ii) Social presence analyses the capacity of the participants to humanise their relationships during an online discussion. It considers how social interactions could be used to model the social climate within a group of learners (i.e., cohesion, affective, and interactive dimensions) (Rourke et al., 1999); (iii) Teaching presence concerns the teaching role before (i.e., course design) and during (i.e., facilitation and direct instruction) the course (Anderson et al., 2001).

Although all three presences contribute to the understanding of learning experience, cognitive presence is the primary element of the CoI model in terms of the operationalisation of the practical inquiry cycle of social-constructivist learning within online environments (Garrison et al., 2001). Within cognitive presence, this cycle is represented in four phases:

1. **Triggering event** initiates the cycle of critical inquiry. In this phase, a student or the instructor proposes a problem or dilemma to trigger the discussion. It could be a problem statement or a question asked by a student;
2. **Exploration** involves an exchange of findings and reflections of initial ideas. Based on the interactions, the participants propose possible solutions for the given problem or explanations for the questions;
3. **Integration** is the phase in which the participants synthesise coherent solutions and knowledge constructed on the ideas and the information shared in the previous phases. In other words, the participants construct structured solutions to the given problem/question by integrating various ideas about the solution;
4. **Resolution** is the phase in which the participants assess the newly-constructed knowledge through hypothesis testing or real-world application in relation to the problem/dilemma that initially triggered the discussion.  

In this study, we focuses on the analysis of cognitive presence since it is the ‘primary issue’ of students’ learning evidence to be explored before the other dimensions (Rourke & Kanuka, 2009). The other two presences of the CoI will be investigated in our future research. Thus, the discussion messages that do not fall into any of the above phases are classified as the Other. Also, the discussion messages that reveal indicators of two phases are classified into the higher one, as the rule of **coding up**. Moreover, examples of student discussion messages explored in this study were provided in **Table 1** for a better understanding of each cognitive phase.

2.2. Content analysis of online discussions with CoI

**Quantitative Content Analysis (QCA)** (Rourke et al., 2001) is broadly applied to assess social and cognitive presences by assigning the units of the online discussion transcripts to a class and counting the number of the units in each class (Bauer, 2007). The CoI model defines three QCA coding schemes, one for each presence. Although the CoI model has extensively been adopted in education research, content analysis has been essentially employed for retrospective analysis, after the courses are finished, without much impact on the actual student learning and outcomes (Strijbos, 2011).

In this regard, the recent literature reports several studies on the adoption of automated methods (i.e., machine learning techniques) to assess CoI presences, intending to drive instructional interventions and enhance student learning outcomes (Kovanović, Gašević, & Hatala, 2014; Rolim et al., 2019). Initially, the methods focused on the use of combinations of traditional text mining features (i.e., bag-of-words vectors) and classification algorithms. For instance, Mcklin (2004) and Kovanović, Joksimović, et al. (2014) proposed the application of a bag-of-words (n-gram) and supervised algorithms based on Neural Networks and Support Vector Machines (SVMs), respectively, to automatically classify online discussion messages into the phases of cognitive presence. The methods reached Cohen’s $κ$ of 0.31 (Mcklin, 2004) and 0.41 (Kovanović, Joksimović, et al., 2014). Moreover, previous works also combined word and document embeddings with LIWC and contextual features to categorise online discussion messages according to the cognitive presence phases (Hayati et al., 2019). Although this study used embeddings to enhance the feature vectors, the paper adopted black-box algorithms in the analysis, such as SVM. The authors did not provide details about the number of messages in the evaluation data, but the best classifier proposed reached Cohen’s $κ$ of 0.68.

The shortcomings of the initial approaches (bag-of-words and black-box classification algorithms) are the lack of: (i) generalisability, implicating that the creation of classification models is dependent on the domain, and (ii) interpretability, which does not provide information on
2.3. Deep learning and explainable artificial intelligence

Deep learning algorithms have gained notoriety due to the outstanding results achieved in computer vision tasks (Russakovsky et al., 2015), but its application rapidly spread to other domains as Natural Language Processing (NLP) (Young et al., 2018). These algorithms rely on the idea that the large number of connections generated on deep neural networks could simulate the same learning process as the human brain (Goodfellow et al., 2016).

In terms of NLP, deep learning has played an increasing role in tasks related to topic modelling/classification (ALRashdi & O’Keefe, 2019) and machine translation (Yang et al., 2020). The Convolutional Neural Network (CNN), as a class of the deep neural networks, has been commonly used for the text classification tasks, especially for small text documents such as Twitter and online discussions posts (Lai et al., 2015; Wei et al., 2017). Some promising studies demonstrated that CNN outperforms traditional machine learning algorithms in this context. For instance, Khamath et al. (2018) compared the results of a deep learning architecture against random forests, SVM, Logistic Regression and Multilayer Perceptron (MLP) in two text classification tasks. The final results showed that the deep learning algorithm achieved results up to 50% better than the other ones. Similarly, Guo et al. (2019) compared different architectures of CNN with Adaboost, a state-of-the-art decision tree algorithm, with the goal of identifying urgency in online discussion messages (Guo et al., 2019). In this context, CNN outperformed Ada-boost by up to 14%.

Despite the promising results, the inability to explain the functionality or reasons of the prediction output from the deep learning algorithms is one of the core barriers that Artificial Intelligence is encountering (Barredo Arrieta et al., 2020). It is also one of the main reasons why the white-box algorithms (e.g. random forests) can be preferable in educational research, notably in the previous studies of the automatic classifier for the cognitive presence (Barbosa et al., 2020; Kovanović et al., 2016; Neto et al., 2018), compared to the deep learning neural networks. However, the field of eXplainable Artificial Intelligence (XAI) aims to create machine learning techniques that enable human audiences to easily understand, reasonably clarify, and appropriately trust the AI methods and implementations (Gunning, 2017). XAI can potentially complement the adoption of deep learning algorithms since it unpacks the model decisions. In learning analytics, explainability of the automatic analysis approach and its product is also a significant goal (Gasevic et al., 2017). It should provide scholars and educators insights into the research questions, for example, the indicators of different cognitive phases revealed in the online discussion messages in our study.

3. Research questions

As discussed in Section 2.1, cognitive presence has a crucial role in analysing knowledge construction and critical thinking in online learning environments. It assesses “the extent to which the participants in any particular configuration of a community of inquiry are able to construct meaning through sustained communication” (Garrison et al., 1999, p. 89). Although several studies proposed automatic methods for classifying discussion transcripts according to the phases of cognitive presence, to our knowledge, no publication applied deep learning techniques for the automatic content analysis of cognitive presence. Hence, our first research question is:

Research Question 1 (RQ1):

To what extent can accurately deep learning methods classify online discussion messages according to the phases of cognitive presence? Can deep learning outperform traditional machine learning techniques for this task?

One of the main limitations reported in the literature is the lack of generalisability of the existing classifiers of cognitive presence (Barbosa et al., 2020; Kovanović et al., 2016; Neto et al., 2018). As deep learning methods are based on the content of the text, it could lead to overfitting of the generated model. Therefore, our second research question is:

Table 1

<table>
<thead>
<tr>
<th>Phase</th>
<th>An example of student discussion messages</th>
<th>Course</th>
</tr>
</thead>
<tbody>
<tr>
<td>Other</td>
<td>Hi XXX I really appreciate your suggestions. I will keep that in mind for future presentations. An interesting start and I think I’m going to enjoy the next 8 weeks.</td>
<td>Software course</td>
</tr>
<tr>
<td>Trigger</td>
<td>I noticed you defined organisational model related more on process actors do you think any other important factors can influence the organisational model study as well? Have you managed to argue the above (that there is no argument) in standard form?</td>
<td>Philosophy MOOC</td>
</tr>
<tr>
<td>Exploration</td>
<td>From my understanding TagViewer transcribes audio and links it to video. Is there future work for the TagViewer to interpret changes in facial responses for example? Even though body language doesn’t have associated audio it does say quite a bit and it would be nice to transcribe that as well.</td>
<td>Software course</td>
</tr>
<tr>
<td>Integration</td>
<td>... I’ve liked using iterative processes as you describe however I don’t believe we should ignore process improvement tools that can be integrated into the process to build better software. I believe the patterns and questionnaires described by the author’s are some of these tools. If we consider usability what you are describing was almost 70% less effective than using the authors patterns ...</td>
<td>Philosophy MOOC</td>
</tr>
<tr>
<td>Resolution</td>
<td>... Agile is by its definition supposed to be flexible and dynamic in nature. As a result I find it ironic that there would be a sticking point on literature Vs real world. I would think the approach should be flexible based on the client and developer skill sets. Software development is a complex human interaction regardless of the project size ...</td>
<td>Software course</td>
</tr>
</tbody>
</table>

why a specific class was chosen for each message (Kovanović et al., 2016). To overcome these problems, the previous methods reported on the literature used domain-independent features and decision trees, especially the random forests algorithm. In this direction, Kovanović et al. (2016), Neto et al. (2018), and Barbosa et al. (2020) adopted random forests and features based on linguistic resources (Coh-metrix (McNamara et al., 2014) and LIWC (Tausczik & Pennebaker, 2010)), latent semantic analysis, named entity recognition, and discussion structures (Waters et al., 2015), to identify the phases of cognitive presence. These studies reached Cohen’s $\kappa$ of 0.63 (Kovanović et al., 2016), 0.72 (Neto et al., 2018), and 0.53 (Barbosa et al., 2020) when applied to discussion messages written in English, Portuguese and across these two languages, respectively.
Research Question 2 (RQ2):

To what extent does the use of deep learning methods affects the generalisability of a classifier for cognitive presence?

Finally, Section 2.2 highlights the need to not only to provide accurate classification of discussion messages but also to explain the reason why messages were classified in one of the cognitive presence phases. Traditionally, deep learning method are categorised as black-box algorithms. However, eXplainable Artificial Intelligence (XAI) (Miller, 2019; Samek & Müller, 2019) is a method to unpack the deep learning model with visual representations of the output. This is a novel approach never adopted in this context. We hypothesise that XAI could provide additional information on the development of cognitive presence phases. Thus, our last research question is:

Research Question 3 (RQ3):

What are the important features revealed from the deep learning with XAI for identifying cognitive presence phases in online discussions? What are the differences between the information provided by deep learning with XAI and previous machine learning methods used in the classification of cognitive presence?

4. Methods

4.1. Data description

In this study, we utilised two data sets coming from the online discussion forums of two different courses. The software engineering data set is from a fully online, masters-level, for-credit course, which was applied in the previous automatic classifier studies (Barbosa et al., 2020; Kovanović et al., 2016). The Logical and Critical Thinking data set is from an introductory Philosophy MOOC, which was used to validate the classification rubric of cognitive presence in the MOOC setting (Hu et al., 2020).

4.1.1. The software engineering course data set

The software engineering course data set (SoftwareSet) consisted of 1747 discussion messages generated by 81 students from six offerings of the course between 2008 and 2011, on the Moodle LMS, at a Canadian public university. The course was a research-intensive course that contained 14 different topics in software engineering, such as software requirements, design, and maintenance. The students were asked to share a video presentation about a course-related research paper, post a new thread in the discussion forum, and invite other students to engage in the debate about the presentation. The students’ participation in the threads accounted for 15% of the final course mark.

All the 1747 messages were manually classified by two expert coders (98.1% agreement, Cohen’s κ of 0.974, which is a largely used measure to evaluate the inter-rater agreement when two coders are involved in the process (Ben-David, 2008; Cohen, 1960)) into five phases of cognitive presence according to Garrison et al.’s coding scheme (Garrison et al., 1999). Both coders are experts in educational technology research with extensive experience with CoI framework. Moreover, one of them has expertise in the course (software engineering), and the other has a background in psychology and education. The disagreements (32 messages) were resolved by discussions of the coders. The third column in Table 2 illustrates the distribution of the cognitive presence phases in SoftwareSet.

4.1.2. The logical and critical thinking MOOC data set

The logical and critical thinking MOOC data set (MOOCSet) was from an archived offering of the Philosophy MOOC, which was designed and taught by a course-design team at a New Zealand’s university, on the FutureLearn platform. This course focused on the basic concepts of logical and critical thinking, how to build good arguments, and how to link the concepts with daily life. The MOOCSet consists of 1917 discussion messages that were randomly selected from the forums under the eight-weekly topics of the archived offering. The offering had 12,311 messages in total generated by 1000 registered learners.

The 1917 messages from the LCT MOOC were manually classified by three trained coders (77.15% agreement, Fleiss’ s κ of 0.763, which is a combination of multiple uses of the Cohen’s κ measure applied when more the two coders participated in the annotation process (Falotico & Quatto, 2015)) based on a revised classification rubric of cognitive presence for the MOOC settings (Hu et al., 2020). This study uses only the messages classified in the same category by all three coders (1479 messages). The forth column in Table 2 displays the distribution of the cognitive presence phases in MOOCSet. The three coders were post-graduate students from the Philosophy Department, who were also the teaching assistants of the LCT MOOC.

4.1.3. Comparison between the two data sets

The discussion messages in both of the data sets share some similarities. One pattern is that the distribution ratio of messages across the four phases of cognitive presence and messages coded as other has a similar trend. For instance, the messages classified into Exploration accounted for the largest proportion, and the Resolution for the least in both data sets. Also, the messages coded as Other, which is called None-cognitive presence in the study of the LCT MOOC set, had the second lowest proportion. Another similarity between the two data sets is that all the messages were contributed by the learners without the instructors’ participation. The learners from both of the courses raised threads in terms of questions and comments around different topics in the videos and articles.

Some differences are also revealed between the two data sets. Firstly, they were from two disciplines (i.e., software engineering and philosophy); therefore, the vocabularies utilised are distinct. Second, the types of the two courses where the messages originate from were different. The software engineering course was a small-scale, for-credit, university course, whereas the LCT MOOC was an open and free course with the participation of the large number of learners. The learners’ demographics and motivations in the MOOC are diverse, which is distinct from the for-credit, university courses.

Moreover, the details of the distribution across the four phases of cognitive presence and Other messages were not exactly the same although the trend was similar. We used the Mann-Whitney U Test to investigate whether the differences of the distribution were significant between the two data sets (McKnight & Najab, 2010). The result (p < .001, U = 1146776, r = 1747&1479, with two-sided hypothesis) indicated that the distributions of the four phases of cognitive presence were statistically different between the data sets. We are aware that these differences could affect the outcomes of our experiments, most notably

<table>
<thead>
<tr>
<th>Research Question 2 (RQ2):</th>
</tr>
</thead>
<tbody>
<tr>
<td>To what extent does the use of deep learning methods affects the generalisability of a classifier for cognitive presence?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Research Question 3 (RQ3):</th>
</tr>
</thead>
<tbody>
<tr>
<td>What are the important features revealed from the deep learning with XAI for identifying cognitive presence phases in online discussions? What are the differences between the information provided by deep learning with XAI and previous machine learning methods used in the classification of cognitive presence?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2: Distribution of cognitive presence phases for both data sets.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>Total</td>
</tr>
</tbody>
</table>

1 In both of our data sets, an individual message refers to a starting post or its replies from each discussion thread.
the between-course generalisability of the automatic classifier for cognitive presence.

4.2. Deep learning architectures

In this study, we used an Inception CNN of naïve form (Szegedy et al., 2015), with the application of pre-trained word-embedding models as word representations to build a classifier for analysing phases of cognitive presence. The architectures of this approach is explained in Section 4.2.2.

4.2.1. Word embeddings

Word embedding is an NLP technique that uses continuous real number vectors to represent texts (i.e. words, phrases, sentences and documents). These vector representations are learned from very large corpora, and are composed of meaningful features associated with positions in a high-dimensional space. The main idea of word embeddings is that semantically similar words or phrases can be located very closely in the space, which also means they have similar representation vectors. All the messages from both data sets were converted into embedding vectors as the inputs for our deep learning neural networks. The main idea of word embeddings is that semantically similar words or phrases can be located very closely in the space, which also means they have similar representation vectors. We used the GloVe pre-trained model (Pennington et al., 2014) to covert the text messages into embedding vectors as the inputs for our deep learning neural networks. For the out-of-vocabulary words (77 words) that GloVe did not contain, we randomly initialised their vectors, which is a common approach in the literature. Clinical data pre-processing involved case-folding and lemmatisation. The max length of tokens for each message was 1939, which contained the complete information for the cognitive presence. We removed the punctuation and numbers in each sentence, and also performed case-folding and lemmatisation. The max length of tokens for each message was 1939, which contained the complete information for all the messages from both data sets. A matrix of shape $D \times L \times N$ was generated as the input variables of our neural networks, in which $\text{dim}_d$ denotes word embedding dimension, $L$ the max length of tokens, and $N$ the number of messages for training.

4.2.2. Convolutional neural networks

The architecture of the naïve-form Inception CNN (Szegedy et al., 2015) used in this study was based on an architecture applied in the classification of discussions on the Quora platform with high performance. Fig. 1 depicts the details of the architecture. It consists of a spatial dropout layer, five convolution layers with max poolings, a batch normalisation process, and a fully-connected layer. Initially, the word embedding matrix, as the vector representation of the input messages, is fed into the spatial dropout neuron with five different filters. Instead of feeding the matrix into the convolution layers directly, the use of the spatial dropout can be effective for preventing neural networks from over-fitting (Lee & Lee, 2020). After the dropout layer, the five convolution layers in one dimension operate the five matrices produced from the dropout layer separately. Then, five max pooling layers retain the important information from the convolution layers but reduce the dimensionality, respectively. After concatenation of the five layers, a new matrix is produced. Next, the new matrix passes through the batch normalisation for improving training speed and validation accuracy before the fully-connected layer computing the prediction probability for each class. Finally, the network outputs one of the five phases of cognitive presence using the activation function of softmax. Table 3 demonstrates the values of the optimal hyper-parameters implemented for all the experiments.

4.3. Explainable artificial intelligence

We are interested in the relevance scores between the input variables and the identified categories on word level. In other words, we wonder which words in the messages are important for each cognitive phase, to understand the deep learning algorithm’s decisions. We applied Gradient-based Sensitivity Analysis (SA), as an example of explainable AI to visualise the importance of words in the discussion messages for identifying the phases of cognitive presence. Similar methods were used to explain deep learning neural networks in NLP tasks (Arras, Horn, et al., 2017,b). Our trained neural network has a prediction function $f_c$ for each category $c$ and an input vector $x$, which consists of dimension $d$ using word embeddings. We use $x_{i,c}$ to denote the $i$th element of the word embedding vector, which represents the $i$th word in a message. Gradient-based SA (Gevrey et al., 2003) assigns the relevance scores $R_{i,t}$ by calculating squared partial derivatives:

$$R_{i,t} = \left( \frac{\partial f_c}{\partial x_{i,t}}(x) \right)^2$$

Theses derivations can be computed by using standard gradient backpropagation (Rumelhart et al., 1986) in most neural networks. SA can also be a decomposition of the squared gradient norm if we sum up the relevance scores of the entire input dimension $d$:

$$\|\nabla f_c(x)\|_2^2 = \sum_d R_{i,t}$$

In this study, we employed the SmoothGrad method from the iNNvestigate library (Alber et al., 2019) as it can effectively improve the gradient-based sensitivity maps by averaging the gradient over number of inputs with added noise (Smilkov et al., 2017). The relevance scores of SA are positive values and were between 0 and 1 in our study. We plotted heatmaps to visualise the word-level relevance scores for each cognitive phase. The words highlighted in red denote strong relevance (0.5 $< R \leq 1$) to a target phase, while words in blues indicate low relevance (0 $< R \leq 0.5$).

4.4. Evaluation metrics

We used the metrics of accuracy, Cohen’s $\kappa$, $F_1$ score, weighted-averaged $F_1$ score, and error rate of each class to measure the performance of our CNN and RF classifiers. Accuracy score and Cohen’s $\kappa$ are commonly used to evaluate the performance of a supervised machine-learning classifier (i.e., input variables have been pre-classified), in educational research (Barbosa et al., 2020; Kovanović et al., 2014b, 2016; Neto et al., 2018; Wang et al., 2015; Waters et al., 2015). Accuracy score is defined as the number of true positives ($TP_c$) plus the number of false positives ($FP_c$) over the sum of all the true and false predictions for each class $c$.

$$\text{accuracy} = \frac{\sum_{c} TP_c + \sum_{c} FP_c}{\sum_{c} TP_c + \sum_{c} FP_c + \sum_{c} FN_c + \sum_{c} TN_c}$$

Cohen’s $\kappa$ (Cohen, 1960) was first introduced to measure the reliability between two human coders on the classification tasks, lessening the agreement caused by chance. It can also be used to evaluate the degree of agreement between the actual and predicted labels in machine learning tasks. Cohen’s $\kappa$ is defined as

$$\kappa = \frac{P_0 - P_e}{1 - P_e}$$

where $P_0$ is the agreement probability of all classes, $P_e$ is the agreement probability by chance, which is computed as the columns and rows marginal probabilities in the confusion matrix (Ben-David, 2008). Macro- and micro-averaged scores are often adopted to compute the overall performance of multi-class classification problems. Macro-averaged method considers all classes as equally important, whereas micro-averaged regards the different contributions of each class, valuing the minority class the least (Van Asch, 2013). Macro-Averaged Precision (macro $AP$) is defined as the mean value of precision score ($P_c$) of each class $c$. 

---

1. The 100-dimension pre-trained vectors were used in our experiments. [Link](http://nlp.stanford.edu/projects/glove/)
4.5. Data processing

To address research question 1, we trained two classifiers based on SoftwareSet (Classifier 1) and MOOCSet (Classifier 2), respectively. The 10-fold cross-validation (CV) method was adopted to randomly divide the entire sample data into 10 non-overlapping folds of approximately equal size. The k-fold CV approach has been commonly used in the machine learning tasks to reduce the risk of overfitting, aiming for more valid result. In every CV loop, the combination of nine-fold data was the training set, and the remainder was the validation set. Moreover, we repeated the training process with 10-fold CV six times by different seeds. Thus, sixty weights file with performance outcomes were generated in order to select the best classifier for identifying the cognitive presence phases by using one of the course data set (SoftwareSet or MOOCSet). The distribution of training and test data in the experiments for research question 1 can be seen in Table 4.

To address research question 2, we used the best weights of the classifiers trained by one dataset to validate on the other one. For example, Classifier 1 was applied to test on the entire MOOCSet (1479 messages), and Classifier 2 on the entire SoftwareSet (1747 messages). Moreover, we trained three new CNN classifiers by concatenation of the two data sets, and evaluated the best models on the partial set of SoftwareSet (Classifier 3), MOOCSet (Classifier 4) and both (Classifier 5). For Classifier 3, the SoftwareSet was split into training and test parts in the ratio of 9:1 by using stratified sampling method. This split before the data concatenation was to ensure that the same distribution of instances from five categories of cognitive presence contained in the training and test sets. After combing the training part (90% of 1747 messages) of the SoftwareSet with the entire MOOCSet, we performed the same training process as in the experiments for research question 1. The model weights with the best performing outcomes was selected to validate on the test part of SoftwareSet (10% of 1747 messages). Similarly, the MOOCSet was split before the concatenation and training process in the

Table 4
Distribution of training and test data in the experiments.

<table>
<thead>
<tr>
<th>RQ</th>
<th>Experiment</th>
<th>Classifier</th>
<th>Training size</th>
<th>Test size</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Train &amp; validate on SoftwareSet</td>
<td>1</td>
<td>1747 (90%)</td>
<td>1747 (10%)</td>
</tr>
<tr>
<td></td>
<td>Train &amp; validate on MOOCSet</td>
<td>2</td>
<td>1479 (90%)</td>
<td>1479 (10%)</td>
</tr>
<tr>
<td>2</td>
<td>Validate Classifier 1 on MOOCSet</td>
<td>1</td>
<td>1479 (90%)</td>
<td>1479 (100%)</td>
</tr>
<tr>
<td></td>
<td>Validate Classifier 2 on SoftwareSet</td>
<td>2</td>
<td>1479 (90%)</td>
<td>1479 (100%)</td>
</tr>
<tr>
<td></td>
<td>Train on both but validate on SoftwareSet</td>
<td>3</td>
<td>3078 (90%)</td>
<td>1479 (100%)</td>
</tr>
<tr>
<td></td>
<td>Train on both but validate on MOOCSet</td>
<td>4</td>
<td>3078 (90%)</td>
<td>1479 (100%)</td>
</tr>
<tr>
<td>5</td>
<td>Train &amp; validate on both</td>
<td>5</td>
<td>2903 (10%)</td>
<td>323 (10% of each set)</td>
</tr>
</tbody>
</table>

---

experiment of Classifier 4. In the final experiment for Classifier 5, we split both of the SoftwareSet and MOOCSet in the ratio 9:1, and then combined each of the 90% data as a new training set, and the 10% as a test set. The details of training-test split of the experiments for research question 2 is also listed in Table 4.

Finally, to address research question 3, we generated several XAI heatmaps to visualise the word-level relevance to every cognitive presence phase by the SmoothGrad sensitivity analysis method. The relevance scores were obtained from the best performing weights and the training data of the experiments.

The source of the experiments in this study is publicly available at github.com/yuanhu19/CognitivePresence_Cnn_XAI repository.

4.6. Random forests classifiers

In order to compare the proposed method, we evaluated the Random Forests (RF) approach derived from the state-of-art model for classification of cognitive presence phases by Kovanić et al. (2016), Neto et al. (2018) and Farrow et al. (2019). We used 199 classification features extracted from the textual information in the two course data sets, 106 features by Coh-metrix (McNamara et al., 2014) and 93 features by LIWC (Linguistic Inquiry Word Count) (Tausczik & Pennebaker, 2010) for English language (Kovanić et al., 2016). In order to maintain the consistency of the classification features with our CNN classifiers, the contextual features, such as the message depth and number of replies and so on, were excluded from the classification features in the RF classifiers. Similarly, the class rebalancing method was not included. The sample data was also pre-processed to remove numbers, and to perform case-folding and lemmatisation, in all of RF classifier approaches.

After the pre-processing and feature extraction, the training-test split in the RF classifiers were exactly the same with the CNN classifiers of each experiment (as shown in Table 4). The 10-fold CV was used to select the best performing RF models by the fine-tuning of two important parameters, ntree (i.e. the number of decision trees built) and mtry (i.e. the number of classification features randomly selected by each tree). To select the optimal ntree value, we examined 100 to 1300 sampled with every 100 interval. For each ntree value, a 10-fold CV was used to perform 30 different numbers, which were randomly selected from 1 to 199, to fine-tune the mtry value. The optimal RF classifiers were created with the best mtry and ntree. In the RF classifier of classifiers for research question 1, a final 10-fold CV was conducted with the entire sample data to report the best performance and SD value. While in the RF classifiers for research question 2, we repeated the training-test loop 10 times with the optimal RF classifiers, as a consistent approach to compare with our CNN classifiers. The optimal parameters in the experiments were: (i) ntree of 900, mtry of 179 in the RF classifiers of Classifier 1, (ii) ntree of 700, mtry of 118 in the RF classifiers of Classifier 2, and (iii) ntree of 500, mtry of 160 in the RF classifiers of Classifier 3, 4 and 5.

5. Results

5.1. Evaluation for individual data set – RQ1

This subsection reports the performance of the CNN classifiers and RF classifiers trained and tested on the individual course data set (SoftwareSet or LCT MOOCSet) for answering the first research question. Table 5 displays the outcome metrics of both the CNN classifiers and the RF classifiers. In the experiment on the SoftwareSet, the best performing CNN Classifier 1 achieved accuracy of 0.632 (SD = 0.03), Cohen’s κ of 0.475 (SD = 0.05), weighted F1 score of 0.649 (SD = 0.03), and macro F1 score of 0.541 (SD = 0.04). The CNN Classifier 2, which was trained and tested on the LCT MOOCSet, showed an overall better performance except the macro F1 score than Classifier 1. Classifier 2 reached an accuracy of 0.707 (SD = 0.03), Cohen’s κ of 0.475 (SD = 0.06), weighted F1 score of 0.723 (SD = 0.03) and macro F1 score of 0.467 (SD = 0.06), in the best case. We can see that the Cohen’s κ coefficient values obtained in both classifiers fell in the ‘moderate’ agreement level (Landis & Koch, 1977). Also, the confusion matrices for both CNN experiments are shown in Figs. 2 and 3. They demonstrate that Classifier 1 (SoftwareSet) had the lowest error rate in the Integration phase, whereas Classifier 2 (MOOCSet) performed the best in the Triggering event phase. Besides, the Exploration phase revealed the second lowest error rate, and the Resolution obtained the highest (i.e. greater than 0.9) in the CNN classifiers for both course data sets.

Compared to the performance of the RF classifiers, CNN Classifier 1 (SoftwareSet) reached similar accuracy, Cohen’s κ and macro F1 scores with minor differences (lower than 0.011). CNN Classifier 2 (MOOCSet) displayed a better Cohen’s κ and slightly lower accuracy and macro F1 scores than the RF classifier 2, with minor differences (lower than 0.025). We can see the weighted F1 scores decreased in both of the CNN classifiers.

5.2. Evaluation across different data sets – RQ2

This section reports the performance of the CNN classifiers that were trained on a course data set but validated on another one for answering the second research question. Table 6 displays the evaluation outcomes of the cross-course experiments. The weights of the best performing classifiers (1 and 2), as reported in Section 5.1, were applied in the first two experiments.

CNN Classifier 1 achieved the accuracy of 0.416, Cohen’s κ of 0.116, weighted F1 of 0.447 and macro F1 of 0.190, when the training set was the SoftwareSet and test set was the MOOCSet. The confusion matrix of this case (Fig. 4) illustrates that the classifier trained on SoftwareSet had lower error rates on the prediction of Other and Exploration in the MOOCSet, and obtained higher error rates on the other three phases. The confusions indicate that many LCT MOOC messages of the Triggering-event and Exploration phases were misclassified into the Other category. Also, many Integration-phase messages from the MOOCSet were misclassified into Exploration phase by the Classifier 1 learned from the SoftwareSet.

CNN Classifier 2, as trained on MOOCSet and tested on the SoftwareSet, reached an accuracy of 0.421, Cohen’s κ of 0.175, weighted F1 of 0.462 and macro F1 of 0.276. It obtained an overall higher performance than the previous experiment. The confusion matrix in Fig. 5 shows that the lowest error rate appeared at the prediction of the Integration phase in the SoftwareSet by using the classifier trained on the MOOCSet. The error rates for identifying the other four categories were greater than 0.65, demonstrating a pool performance. Moreover, the majority of the misclassification occurred between the Exploration and Integration phase.

Table 5

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Accuracy (SD)</th>
<th>Kappa (SD)</th>
<th>Weighted F1 (SD)</th>
<th>Macro F1 (SD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classifier 1 Train &amp; test on the SoftwareSet (CNN)</td>
<td>0.632 (0.03)</td>
<td>0.475 (0.05)</td>
<td>0.649 (0.03)</td>
<td>0.541 (0.04)</td>
</tr>
<tr>
<td>Classifier 2 Train &amp; test on the LCT MOOCSet (CNN)</td>
<td>0.707 (0.03)</td>
<td>0.528 (0.06)</td>
<td>0.723 (0.03)</td>
<td>0.467 (0.06)</td>
</tr>
<tr>
<td>RF classifier 1 for RQ1 Train &amp; test on the SoftwareSet (RF)</td>
<td>0.642 (0.04)</td>
<td>0.479 (0.05)</td>
<td>0.667 (0.05)</td>
<td>0.530 (0.03)</td>
</tr>
<tr>
<td>RF classifier 2 for RQ1 Train &amp; test on the LCT MOOCSet (RF)</td>
<td>0.720 (0.04)</td>
<td>0.488 (0.08)</td>
<td>0.751 (0.05)</td>
<td>0.487 (0.06)</td>
</tr>
</tbody>
</table>
In both of the cross-course experiments, the Cohen’s κ values were at a ‘slight-agreement’ level (Landis & Koch, 1977). Also, both of them reached a zero accuracy for the classification of the Resolution phase. Outcomes of the RF classifiers for both experiments were also listed in Table 6. The CNN classifier trained on the MOOCSet and tested on the SoftwareSet had a better performance than the RF classifier with the same condition, whereas the CNN classifier reached slightly lower outcomes than the RF approach for the other way around (i.e., trained on the SoftwareSet and tested on the MOOCSet).

![Fig. 2. Confusion Matrix by training and testing on SoftwareSet.](image1)

![Fig. 3. Confusion Matrix by training and testing on LCT MOOCSet.](image2)

Table 6
Outcomes of the CNN classifiers for cross-course validation compared to the RF classifiers (RFs).

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Accuracy</th>
<th>Kappa</th>
<th>Weighted F1</th>
<th>Macro F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classifier 1 Train on SoftwareSet but test on MOOCSet (CNN)</td>
<td>0.416</td>
<td>0.116</td>
<td>0.447</td>
<td>0.190</td>
</tr>
<tr>
<td>Classifier 2 Train on MOOCSet but test on SoftwareSet (CNN)</td>
<td>0.421</td>
<td>0.175</td>
<td>0.462</td>
<td>0.276</td>
</tr>
<tr>
<td>RF classifier 1 for RQ2 Train on SoftwareSet but test on MOOCSet (RF)</td>
<td>0.454</td>
<td>0.188</td>
<td>0.453</td>
<td>0.262</td>
</tr>
<tr>
<td>RF classifier 2 for RQ2 Train on MOOCSet but test on SoftwareSet (RF)</td>
<td>0.419</td>
<td>0.079</td>
<td>0.512</td>
<td>0.206</td>
</tr>
</tbody>
</table>

We also examined the performance of the CNN classifiers trained on the combination of the two course data sets by three experiments. Table 7 displays the outcomes of the CNN classifiers trained on the combined set and tested on partial SoftwareSet, MOOCSet and both, respectively. We can see that Classifier 4 performed better than Classifiers 3 and 5. Our CNN classifiers of the combined-course experiments had lower performance than the RF classifiers. Moreover, the overall results of the combined-course experiments surpassed the outcomes of the cross-course validations (Table 6), but were lower than the results in the individual-course experiments (Table 5).

5.3. Visualisation by the explainable AI – RQ3

The results of the XAI heatmaps were generated from the experiments of the best performing classifiers trained and tested on the individual course data set, SoftwareSet or LCT MOOCSet. These two classifiers can demonstrate purer comparative outcomes of the word-level predictive indicators of the cognitive presence phases between the two data sets, than the classifiers trained in the combined-course experiments. Due to the page limitation, we provided four messages classified into Exploration (Fig. 6) or Integration (Fig. 7) phase from both the data sets, as the examples of the XAI visualisations.

We can observe that the word constructs and accommodate were highly relevant with classification of the Exploration phase in SoftwareSet, followed by the programming methodology and function as moderately positive (Fig. 6a). While in MOOCSet (Fig. 6b), the word conclusion was revealed a high relevance to identify the Exploration phase, and the words agree, example, even, premises and true performed moderate
relevance. In comparison with the positive associations, the majority of the lowest relevant words in both the data sets were ‘stop-words’, which refers to the most commonly used words in a language, such as the, to, a, is in Fig. 6. Apart from the stop words, there were some nouns highlighted as low relevance to the Exploration phase in the SoftwareSet message (Fig. 6a), namely programmer and adaptation.

We also provided two sample messages classified into the Integration phase from the two data sets in Fig. 7. In the message from the SoftwareSet (Fig. 7a), the highly relevant words for classification of the Integration phase contained implementation, i, fold, and long, and the moderately relevant words included corresponding, exception, aspects, and methodology. While the low-relevance words were presents, actually, reports, and however and also the stop-words. Comparatively in the MOOCSet message (Fig. 7b), the words that indicate strong relevance were correlation, between, associated, relation, lateralisation, and functional, whereas the low-relevance words were divergent, necessarily, hemisphere(s), specifically, cite and so on.

6. Discussion

6.1. The performance comparison between our CNN classifiers and RF baselines – RQ1

The CNN classifiers presented in this study reached Cohen’s $\kappa$ of 0.475 and 0.528 for the SoftwareSet and LCT MOOCset, respectively, which demonstrates a moderate inter-rater agreement (Landis & Koch, 1977). These results were similar to the performance of the RF classifiers which used random forests with LIWC and Coh-Metrix features, and the outcomes of the experiments without the over-sampling methods reported in the previous studies reported on the literature (Kovanovi´c et al., 2016; Farrow et al., 2019; Barbosa et al., 2020). In addition, the CNN classifiers obtained a smaller error rate in the Integration phase (Figs. 4 and 5) compared to the previous works (Barbosa et al., 2020; Farrow et al., 2019). Conceptually, the Integration phase can be characterised by a knowledge construction process where the students formulate new concepts and solutions (Garrison et al., 2001). The error reduction in this phase could be related to the adoption of pre-trained word embeddings, which include an extensive vocabulary compared to the traditional machine learning methods (Pennington et al., 2014).

6.2. Generalisability of our CNN classifiers across different data sets – RQ2

Another important finding of this study concerns the analysis of the generalisability of a method based on textual-related features. While the literature suggests that the adoption of linguistic resources (i.e., LIWC and Coh-Metrix) and structural information (i.e., discussion context features) (Kovanovi´c et al., 2016; Farrow et al., 2020; Barbosa et al., 2020) can increase the generalisability of a classifier, our results demonstrate that the use of the CNN classifiers on cross-course validations can reach similar results (Table 6). This finding implies that the deep learning approaches with word embeddings can have potential to increase the generalisability if we can build neural networks including the sequential structures of the discussion messages, such as Recurrent Neural Networks (Graves et al., 2013) and Transformers (Wolf et al., 2020), in future work. Also, the generalisability could be improved by using the advanced pre-trained models (e.g. BERT (Devlin et al., 2018)) and fine-tuning them with the corpora that contain the domain-specific vocabularies in the sample data sets.

Random forests classifiers achieved better results when the training step was performed in the combined data set (the SoftwareSet and LCT MOOCSet). A possible reason for this could be found in the fact that the optimal parameters (i.e. $ntree$ and $mtry$) were fine-tuned separately in the RF classifiers for the individual-course and combined-course experiments. However, we used the same hyper-parameters, such as the number of filters, batch sizes and kernel sizes, in all of our CNN experiments to maintain the consistency. This result provides us a hint that the hyper-parameters should be adjusted according to the change of the data set.

6.3. Important features revealed from the explainable AI – RQ3

This paper also introduced some examples of the eXplainable Artificial Intelligence (XAI) visualisations to investigate the indicators for

![Fig. 4. Confusion Matrix with SoftwareSet as the training set and MOOCSet as the test set.](image-url)
the phases of cognitive presence through the training of deep neural networks. Interestingly, the first four high relevant words we mentioned in the MOOCSet message (Fig. 7b), and the word corresponding in the SoftwareSet message (Fig. 7a), can be the vocabularies for ‘connecting ideas’ and ‘convergence’, which are the important indicators to identify the Integration phase in the coding scheme of cognitive presence (Garrison et al., 2001; Hu et al., 2020). Besides, most the high-relevance words in the examples of the Exploration phases (Figs. 6 and 7) were the course-relevant concepts previously used in the discussions, which is align to the definition of this phase (Garrison et al., 2001). These findings suggest that the use of expressions, which indicate relationships between different subjects, ideas or arguments in a discussion message, can be significant features to identify the Integration phase. Also, in shorter messages, the use of specific nouns or noun phrases, which are the core concepts delivered in the courses, could be important indicators to identify the Exploration phase.

We also found the important features provided by our deep learning with XAI method were word-level based, which differs from what the previous works have found by the Mean Decrease Gini (MDG) interpretation (Barbosa et al., 2020; Kovanović et al., 2016). The previous works proposed more abstract features, such as the degree of lexical diversity, the degree of meaningfulness content words and so on. Educational researchers and course instructors have to seek the expressions matched with definitions of those abstract features in the Coh-Metrix and LIWC tools for each message. This work is still time-consuming and nontransparent. In comparison, the word-level explanations of XAI are perceivable and understandable. Nevertheless, these two explainable methods have some similarities. For instance, Kovanović et al. (2016) suggests that messages, which had higher semantic similarities with their previous ones, tend to be more relevant to the higher phases (i.e., Integration and Resolution). This information could be connected with the results in our XAI visualisations as it reflects that the expressions in terms of associating ideas in the messages can be strong indicators for the Integration phase. Similarly, the XAI could shine a light on the topic of a question asked during the discussion since the number of questions marks can be strong indicators of Triggering event messages according to the previous studies (Barbosa et al., 2020; Kovanović et al., 2016). A divergence from the previous work is also reflected. Several important indicators proposed in the previous work were related to the stop-words (i.e., number of personal pronouns, prepositions, indefinite pronouns, pronouns in the third person singular); however, the XAI highlighted many of them as low-relevance indicators.

Therefore, we suggest that the deep learning method (CNN and XAI), and the previous approach (random forests and MDG (Barbosa et al., 2020; Farrow et al., 2020; Kovanović et al., 2016)) could be applied as complementary approaches. On the one hand, it is possible to propose a compound classifier to benefit from the best predictive power of each approach since random forests better predicts the Triggering event and the Exploration phases, and CNN obtained more favourable results for the Integration phase. On the other hand, XAI could provide additional insights (i.e., word-level positive and negative indicators) on the most relevant features extracted by MDG.

7. Final remarks

This paper has three contributions. First, the introduction of a deep learning model to automatically classify online discussion messages into cognitive presence phases. The results reached Cohen’s κ values of 0.498, a moderate inter-rater agreement. Second, the proposed
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References


Y. Hu et al.

Computers and Education: Artificial Intelligence 2 (2021) 100037


