Student-generated texts as features for predicting learning from video lectures: An initial evaluation

Ilias Karasavvidis, Charalampos Papadimas, Vasiliki Ragazou
ikaras@uth.gr, papadimas@uth.gr; ragazou@uth.gr

Department of Early Childhood Education, University of Thessaly, Greece

Abstract

The digital trails that students leave behind on e-learning environments have attracted considerable attention in the past decade. Typically, some of these traces involve the production of different kinds of texts. While students routinely produce a bulk of texts in online learning settings, the potential of such linguistic features has not been systematically explored. This paper introduces a novel approach that involves using student-generated texts for predicting performance after viewing short video lectures. Forty-two undergraduates viewed six video lectures and were asked to write short summaries for each one. Five combinations of features that were extracted from these summaries were used to train eight machine learning classifiers. The findings indicated that the raw text feature set achieved higher average classification accuracy in two video lectures, while the combined feature set whose dimensionality had been reduced resulted in higher classification accuracy in two other video lectures. The findings also indicated that the Gradient Boost, AdaBoost and Random Forest classifiers achieved high average performance in half of the video lectures. The study findings suggest that student-produced texts are a very promising source of features for predicting student performance when learning from short video lectures.

Keywords: machine learning, raw text features, engineered text features, video lectures, video learning analytics

Introduction

Nowadays, e-learning is ubiquitous, its adoption ranging from corporations to educational institutions. In the context of higher education both hybrid and fully online courses have become the norm. When studying online, students engage in various course activities such as lecture viewing, online readings, synchronous and asynchronous online discussions, quizzes, assessments, and assignment submissions (Ferguson, 2012; Hernández-García & Conde-González, 2016; Romero & Ventura, 2010; Tomasevic, Gvozdenovic & Vranes, 2020). All student activity in such E-learning systems can be logged. Hence, students' interactions with the course materials are reflected in their digital footprints (Ifenthaler & Widanapathirana, 2014; Papamitsiou & Economides, 2014; Romero & Ventura, 2010; Schumacher & Ifenthaler, 2018). Through their participation in e-learning environments, students generate huge amounts of digital data. The digital traces that the students leave have attracted the interest of educators, researchers, administrators, and other stakeholders.

Over the past decade, Educational Data Mining (EDM) and Learning Analytics (LA) emerged as research fields that aim to capitalize on such data (Ferguson, 2012). According to Romero and Ventura (2020), LA can be defined as the measurement, collection, analysis, and reporting of data about learners and their contexts, for purposes of understanding and optimizing learning and the environments in which it occurs. Video Learning Analytics (VLA) is a sub-set of LA that focuses on students' learning behaviours when learning from video lectures (Giannakos, Chorianopoulos & Chrisochoides, 2014; Hasan et al., 2020).

The value of the data that is collected through online learning systems is widely recognized. The first wave of studies in the field focused on the identification of variables that could be used to predict student performance, increase student retention, and identify at risk students.

More specifically, LA studies have employed various measures such as (a) demographic data: age, date of birth, citizenship, prior academic history, expertise (Stöhr, Stathakarou, Mueller, Nifakos & McGrath, 2019; Tempelaar, Rienties, Mittelmeier & Nguyen, 2018) native language, previous enrolment in the same course (Gašević, Dawson, Rogers & Gasevic, 2016), (b) engagement or interaction data: student online forum discussions (Chen, Chang, Ouyang & Zhou, 2018), clickstreams (Giannakos et al., 2015; Stöhr et al., 2019), trace data recorded by LMSs e.g. course logins, resources, assignments, quizzes, feedback, map, lessons, and chat) (Agudo-Peregrina, Hernández-García, & Pascual-Miguel, 2014), and (c) performance data: final exam (Kim, Yoon, Jo & Branch, 2018), midterm exams (Blikstein et al., 2014), and online quizzes (Huang, Lu, Huang, Yin & Yang, 2020).

By drawing on the aforementioned features former studies have been able to predict the final course performance (Agudo-Peregrina et al., 2014; Conijn, Snijders, Kleingeld, & Matzat, 2017; Tomasevic et al., 2020) or identify students who are about to drop out due to learning struggles (Hasan et al., 2020; Huang et al., 2020). Overall, the picture that emerged from utilizing such data is very promising indicating that student performance can be modelled using such features to meet the needs of instructors, students, and administrators.

Unlike other types of features, student-produced texts have not been systematically explored as features for predicting student performance, either in E-learning or in micro-learning contexts. This is somewhat paradoxical considering that students generate a large volume of texts when interacting with E-learning systems. The only notable exception to this pattern is forum discussions. In fact, participation in forum discussions has turned out to be an important predictor of online learning success (Papamitsiou & Economides, 2014; Schumacher & Ifenthaler, 2018). Notwithstanding the use of text, most studies tend to consider behavioural and contextual features (e.g., number of words per forum post, time of post etc.) rather than the actual words used and their semantic meaning.

The present study aims to address this limitation by explicitly focusing on student-produced texts. More specifically, we focus on a cohort of students who viewed a series of video lectures and were required to draft short summaries for each lecture. Using Natural Language Processing (NLP) techniques we extract a combination of five different sets of features and use them to train eight Machine Learning (ML) algorithms. Our goal is to explore the potential of text as a source of features for predicting student learning from video lectures.

The paper is organized as follows. The 'Literature review' section outlines former studies in this field. The 'Method' section introduces five text-based feature sets: (a) raw text features, (b) engineered features, (c) a combination of engineered features and raw features, (d) a combination of all features whose dimensionality has been reduced and (e) a combination of all features with feature reduction applied to engineered features. The 'Analysis' section describes the NLP techniques for text processing and the ML algorithms that were utilised. The 'Results' section presents the performance of different ML classifiers as a function of the input feature sets. The 'Discussion' section contextualizes the paper findings and explores the implications and limitations of this study. Finally, the 'Conclusion' section summarizes the major key points of the paper.

Literature Review

Text as a source of features for modeling student performance

As mentioned above, student-produced texts are not commonly used as features for predicting learning performance or modelling student learning. Still, student-generated texts have attracted research attention. Of particular interest in the context of online learning are texts created in fora, chat, Wikis, and assignments that require the submission of essays.

Earlier research had indicated that forum discussions are correlated with learning performance (Kim, Park, Yoon & Jo, 2016; Papamitsiou & Economides, 2014). Hence, it was only natural that researchers explored the power of forum discussions as features for predicting learning (Conijn et al., 2017; Kim et al., 2016; Romero & Ventura, 2010; Wang, Kraut & Levine, 2015).

Interestingly, most studies that have examined texts as a feature either for predicting performance or other reasons are related to MOOCs (Robinson, Yeomans, Reich, Hulleman & Gehlbach, 2016). The motivation behind such studies lies in the inability of instructors to follow the online discussions that take place in forums and intervene as needed. The sheer volume of posts in MOOCs makes it practically impossible for a single instructor or even a small, dedicated course team to keep track. Simply put, monitoring MOOC forum discussions is a formidable task. As a rule, such large-scale courses call for automation. To date, some studies have explored automated tools that could help instructors identify which topics are unrelated to the course content (Wise, Cui, Jin & Vytasek, 2017; Wise, Cui & Vytasek, 2016), detect posts that reflect cognitive presence (Hayati, Chanaa, Idrissi & Bennani, 2019) or find posts suggesting that an urgent response is required (Almatrafi, Johri & Rangwala, 2018). It should be noted that the massive number of messages that is readily available in MOOCs provides researchers with sufficiently large text corpora that can be used for training ML models.

Former studies that have utilized texts as a source of features for ML can be distinguished in two large categories: (a) studies adopting non-linguistic features and (b) studies utilizing linguistic features.

With respect to the former, the literature indicates that a series of studies have systematically examined texts but without considering their linguistic or semantic content, i.e., no NLP techniques are used. More specifically, many studies that use course forum discussions as a source of features do not extract any linguistic features from the fora whatsoever (Romero, López, Luna & Ventura, 2013; Rossi & Gnawali, 2014; Wen, Yang & Rose, 2014). Studies falling into this category typically utilize contextual information such as the number of posts, the number of words per post or the time of posting.

For example, in a study involving 114 undergraduates in a CS course, Romero et al. (2013) extracted – among others - the following features from the course forum: messages, threads, words, sentences, views, and time. Their aim was to use these features to train 14 ML algorithms to predict whether students would pass or fail the course. Interestingly, the number of messages and the number of words per message were amongst the best predictors, yielding a classification accuracy that ranged from 0.70 to 0.90.

Rossi and Gnawali (2014) examined a huge corpus of messages (more than 700K posts) from 60 Coursera MOOCs using mostly non-linguistic features. In fact, word count was the only linguistically motivated variable that they used. They adopted a language-independent approach for supervised learning that involved the classification of threads into Coursera's labelling system (6 general categories of threads: Study Groups, General Discussions, Lectures, Assignments, Logistics, and Course Material). The authors reported a classification accuracy that ranged from 0.58 to 0.91, though one specific category (Study Groups) was easier to classify more accurately compared to all others. Interestingly, the average number of words per message, namely the only NLP feature they had considered, was amongst the most important predictors.

Despite the promising findings of both studies' classification-wise, two points are worth noting. First, linguistic features such as the number of words turned out to carry important information. Second, the features extracted from the forum texts were not semantic in nature, which means that the actual information that the words carry is ignored, which is a major limitation of such approaches.

With respect to the latter, a handful of studies have explicitly focused on linguistic or semantic features of student-produced texts of various forms. As a rule, the NLP methods used involve text vectorization either through Bag of Words (term frequencies) or distributed word representations (word embeddings). Some studies typically adopt the Bag of Words (BoW) approach (unigrams and bigrams) often in combination with TF-IDF (Cui & Wise, 2015; Wise et al., 2017; Wu, Hsiao & Nian, 2020; Yang, Wen, Howley, Kraut & Rose, 2015). Other studies have also explored Part of Speech (POS) tagging as features (Lin et al., 2009). With a few notable exceptions (e.g., Dessì, Fenu, Marras & Reforgiato Recupero, 2019; Zhang & Litman, 2018), word or sentence embeddings are hardly ever used. Considering the objective of this study, we review studies that use NLP in the next section.

Extracting features from text for training ML models

The literature search returned a handful of studies that have employed NLP techniques for analysing student-produced text. Robinson et al. (2016) used data from a HarvardX MOOC focusing on Education with more than 41K participants. In particular, they analysed student written responses to a pre-course survey. This open-ended prompt asked students to identify why the course is useful or how they would apply the knowledge and skills that they expect to gain from the course. The objective was to use NLP to predict who completed the course or not (i.e., binary classification). Student responses were vectorized using e.g., BoW approach involving unigrams and bigrams. Subsequently, these vectors were used to train a Logistic Regression model. The classification accuracy that was obtained using a combination of NLP and other features was somewhat promising (0.59).

In a study that targeted discussion forums in five different MOOC courses (3 of which were Statistics-related), Wise et al. (2017) examined the possibility of automatic classification of posts as content-related or unrelated (binary classification). The researchers sampled a large corpus of messages from the course fora and vectorized the texts using a BoW approach with unigrams and bigrams. A total of 2.2K features that were extracted were used to train a Logistic Regression model. The study results indicated that the model accuracy was 0.80 and that it could generalize both to other Statistics courses and even to an unrelated course - though the accuracy dropped to 0.73.

Almatrafi et al. (2018) used the Stanford MOOCPosts dataset, which involves more than 29K posts in three domains (Humanities, Medicine, and Education), to detect whether a MOOC forum post requires an urgent instructor response or not. For this binary classification task, they used three feature sets: LIWC (94 features), post metadata (number of reads, number of votes, first post/comment), and NLP (BoW with unigrams). Various combinations of these feature sets were used to train several ML classifiers: Naïve Bayes, Support vector machines, Random forests, AdaBoost, and Logistic regression. The authors report that the performance of Logistic Regression with unigrams was 0.84.

Dessì et al. (2019) collected more than 10K video lectures from 617 Coursera courses to classify educational videos in seven general-level categories. The authors extracted the transcripts from the video lectures and then sought to classify the videos using a combination of features. More specifically, for this multi-class classification problem they utilized three NLP feature sets: (a) a BoW approach with unigrams and TF-IDF, (b) keywords and (c) concepts. It should be noted that for (b) and (c) the authors used IBM Watson's Natural Language Understanding API. Similarly, to the BoW TF-IDF, IBM Watson's Features Extraction Module converts both keywords and concepts into vectors. These feature sets were used to train five ML classifiers: Decision Tree, Support Vector Machine, Random Forest, and Support Vector Machine with Stochastic Gradient Descent. The results indicated that the F-measure was around 0.70 when the TF-IDF was used as a feature and SVM or SVM+SGD were used as algorithms.

Rationale of the present study

Overall, the aforementioned studies reveal the potential of student-authored texts as well as other texts (e.g., video transcripts or written responses to pre-course surveys) as ML features. While all these studies involved supervised learning in either binary or multi-class classifications, to the best of our knowledge no study has explicitly targeted learning performance when learning from video lectures. The present study aims to fill this gap by exploring NLP techniques for predicting learning performance in micro-learning videos.

The main complication of using student-generated texts for ML purposes is that text is not numerical data, so it cannot be directly used. Hence, text vectorization is an essential prerequisite: words need to be converted to numbers, which could then be used as features for training ML models. In the next sub-section, we introduce the essential NLP terminology that will be used throughout this paper.

Converting Text to Vectors for ML

Document: In NLP a document is a sequence of words, ranging from a few words in a sentence, a paragraph and even to longer text chunks such as a whole article. It should be noted that typically each word in a document is a feature on its own.

Document Term Matrix (DTM): In NLP the DTM is commonly used to represent text in a numeric structured form. In such a matrix, the rows represent the documents while the columns represent the individual words. The elements of such a matrix are binary numbers (one-hot encoding), word frequencies (BoW) or weighted word frequencies (TF-IDF). The length of the matrix (i.e., columns) is determined by the vocabulary over all documents.

One-hot encoding: In this case the words contained in a document are represented as binary numbers. Hence, if a word is present in the document the value in the corresponding column in the DTM is 1, if not it is 0. It should be noted that only the presence of a word in a document is considered, i.e., its frequency is ignored.

BoW: In this case texts are represented as fixed-length feature vectors. Unlike the one-hot encoding approach, the corresponding value in the DTM column represents the frequency of a given word. The frequency of words reflects their importance: words with higher frequency of occurrence are given more importance relative to other words.

TF-IDF: While the TF-IDF approach also uses word frequencies, these are weighted, which addresses the fact that some words appear more frequently than others. In this case, higher weights are given to words that are particular to a specific document, namely have less frequency over all documents. Each column in the DTM contains a value that corresponds to the weighted frequency of the word.

Word embeddings: There are three main limitations associated with the representation of words in frequency-based approaches such as BoW and TF-IDF. First, a large vocabulary will entail that the total number of dimensions will be high, which will result in a very sparse matrix where most elements will be zero. Second, the computation of similarity between two documents presupposes the occurrence of common words. If there are no common words in the documents, then the resulting cosine similarity will be zero even if the two documents have identical meanings. Last, the BoW and TF-IDF only consider the presence of words, that is they do not capture the meaning of words. Word embeddings address these limitations. More specifically, word embeddings are high-dimensional vectors representing words. Word embeddings are trained by examining the contexts in which a word occurs. In particular, a neural network is used to predict the word using a specific window size commonly using two architectures: Continuous Bag of Words and Continuous Skip-gram (Mikolov, Chen, Corrado & Dean, 2013). Each word is represented as a dense vector of n-dimensions. The values of each vector are the weights of the hidden layers of the neural network. What is important in the case of word embeddings is that they capture the semantic meaning of words in a number of different dimensions.

Rationale behind feature selection

As it was pointed out above, we have been unable to find any studies that utilize text features for predicting student performance in e-learning settings, particularly when learning from video lectures. Considering the bulk of texts that students generate in online courses (e.g., forum posts, instant messages, written assignments etc.), it is interesting to note that the potential of using such texts as features has not been exploited. More specifically, while previous studies have actually examined text-based features, their main limitation is that they have commonly employed frequency-based approaches (BoW, TF-IDF). As it was mentioned above, this means that these approaches do not capture the semantic meaning of the words. For example, it has been established that participation in online forum discussions is correlated with course performance. Specifically, Romero et al. (2013) found that the number of messages and the number of words per message were amongst the strongest predictors of whether students would pass or fail a course. While this is undoubtedly an interesting finding, features such as the sheer number of forum posts do not necessarily reflect student understanding because the semantic content of the posts is ignored. The same applies to features such as the number of words per post: this figure does not reflect students' understanding whatsoever. Features such as the number of posts or the number of words per post entail that the semantic information that the words carry is ignored.

Compared to former studies, the approach introduced in this work differs in three main respects: (a) performance prediction, (b) similarity measures, and (c) reference frame.

First, previous studies have employed features extracted from student-generated texts for various classification tasks: predict whether students would pass or fail the course (Robinson et al., 2016), identify forum posts as related or unrelated to the discussion topic (Wise et al., 2017); determine if a forum post requires urgent instructor attention or not (Almatrafi et al., 2018); and classify educational videos in terms of topic (Dessì et al., 2019). Our contribution lies in that we have extracted text features and attempted to predict the comprehension of video lectures, which is something that has not been systematically explored before.

Second, with one exception (Dessì et al., 2019), previous research has largely utilized frequency-based text vectorization techniques (e.g., BoW approaches with TF-IDF with unigrams and bi-grams). While promising, these techniques have certain shortcomings – some of which have been already mentioned (e.g., sparse matrices; misleading cosine similarity if there are no shared words; only the presence of words is captured, not their meanings). Word embeddings have the potential to capture the semantic content and have been generally utilized as an input to machine learning classifiers, which empowers machine learning methods to scrutinize unstructured text (Wang et al., 2018). Our innovation lies in that we have used word embeddings as features rather than simply one-hot encoding or frequency-based vectorization techniques. Thus, in addition to standard features that are commonly used in the NLP literature, we have utilized word embeddings.

Third, we have also introduced an additional innovation: the use of video lectures as a point of reference. More specifically, we have used the video lecture transcripts as a reference against which student summaries were compared. We considered the video lectures to represent the ground truth and these were the yardstick against which the student summaries of the video lectures were evaluated. Both the video transcript and each student summary are treated as documents (see aforementioned definition). Using cosine similarity (i.e., the dot product of two-word vectors) we have attempted to determine how well student summaries captured the essence of each video lecture.

The main working assumption is that the higher the similarity between the two documents the higher the probability that the student has captured the essence of the video. Consequently, it is also highly likely that that the student has correctly understood the underlying concepts. Based on Learning Sciences research we assume that the study of the summaries students produce is a direct and dynamic indicator of their understanding of the concepts covered in a video lecture. One the one hand, texts have been extensively used in cognitive psychology for representing student understanding (e.g., Kintsch, 1988) On the other hand, according to the generative activity principle of Multimedia Learning Theory (Mayer, Fiorella & Stull, 2020), learning from instructional videos is facilitated when learners engage in active processing of the content such as summarizations or explanations of the materials. It should be noted that while other studies have also focused on video transcripts (Atapattu & Falkner, 2018; Dessì et al., 2019), their motivation has been entirely different. We treat the semantic similarity between a student summary and the corresponding video lecture transcript as indicative of student's understanding of the lecture contents. Therefore, our contribution involves the similarity check of the short summaries that the students generated against the video lecture transcripts using word embeddings.

Against this background, the present study examined two main research questions:

RQ1: Which set of text features yields the highest classification accuracy?

RQ2: Which ML algorithms yield the highest classification accuracy?

Method

Research design and Data collection

The data reported in this paper originated from a large, ongoing research project which adopts a design-based research approach (Cobb et al., 2003; Collins et al., 2004) that is tailored to EDM and LA (Rienties, Cross & Zdrahal, 2017).

This study recruited 42 undergraduate students (40 females and 2 males; age: M = 21.25, SD = 2.5) from a higher educational institution in Greece. The majority of the participants were first-semester students (83.33%), while the rest came from seventh (4.76%) and ninth semesters (11.90%). Based on an initial survey, more than 75% percent of the participants reported that were very familiar with media software, office, internet, and social media applications. Moreover, all participants reported access to a computer at home and a high-speed Internet connection. Participation in the study was voluntary and the students were compensated with two-course credit points. Formal approval for the study was obtained from the Ethical Committee of the University.

LMS

Moodle was the LMS that was adapted and customized for the purposes of this study. All student activity (e.g., viewing the video lectures, responding to quizzes and questionnaires) was thoroughly recorded.

Video lectures

Six video lectures were developed in vitro which covered topics related to digital media, using an introductory textbook as a source (Manovich, 2013). Table 1 presents the titles, the topics, and the duration of each video lecture.

Table 1. Topics of the video lectures

Video Lecture (id)	Title	Duration (min: sec)	Topics
1	Digital media	9:20	Introduction to digital media
2	Simulation	10:27	Phase 1: Simulating analog media in digital systems
3	Hybridization	10:29	Phase 2: Creation of media hybrids
4	Deep remixability	12:06	Phase 3: Deep remixability
5	Digital compositing (part 1)	8:57	The role of transparency in compositing Types of layers in images Examples of 2D and 3D compositing effects
6	Digital compositing (part 2)	8:19	The structure of digital image 2D digital compositing example 3D digital compositing example

Measures

Summaries: After viewing each video lecture, the participants were asked to write a short summary of the main concepts covered. Their responses were typed in and submitted through the LMS.

Knowledge tests: We developed three types of knowledge tests for this study: a pre-test, six comprehension tests, and a post-test. The pre-test comprised 14 closed type items. This test assessed students' prior knowledge on multimedia topics (Examples: "A digital image is made up from pixels (True – False), "The term digital composition refers to the combination of at least two images to create a new single image" (True-False). Second, six comprehension tests were developed to measure students' conceptual knowledge for each video lecture (Examples: "Social media is an example of hybrid media": True – False"). Each knowledge test comprised ten closed type items. Third, the post-test comprised 16 closed type items (True-False, Multiple Choice). Examples of post-test items: "Transparency is now so prevalent that it is integrated into virtually any type of software involving visual media (e.g., word processor, browser, etc.)" (True-False), "Instagram is a case of hybrid media" (True-False). In all aforementioned measures 1 point was given for correct responses and 0 points for incorrect ones. Consequently, the maximum obtainable scores for the pre-test, each comprehension test, and post-test were 14, 10, and 16 respectively.

Procedure

Due to pandemic lockdowns, the study was eventually conducted online, and a considerable number of students who had initially expressed interest did not eventually participate. The total study duration was 3 hrs. Prior to the treatment the first author had extensively briefed the participants about the study objective, conditions, and procedures. The subjects were asked to follow a specific learning path on the LMS that consisted of three parts. First, the students filled a demographics survey and completed the pre-test. Second, they watched a series of six video lectures. Following each viewing they were asked to write a short summary of the topics covered in the respective lecture. It should be noted that the students were explicitly requested to refrain from using any resources such as memory aids or note-keeping during the video lectures. Their responses were to rely exclusively on the information they had committed to memory while watching the video lectures. Next, they answered a quiz targeting memory recall and comprehension. This process (viewing, summary, and quiz) was repeated for the remaining video lectures. The treatment was concluded with the administration of the post-test.

Analysis

The study adopted the Python ML ecosystem and specifically the Pandas, Scikit-Learn (Pedregosa, Varoquaux, Gramfort, Michel & Thirion, 2011) and spaCy libraries (Explosion, 2022). Fig.1 summarizes the overall data workflow adopted for the study.

The procedure of speech to text conversion was performed with Vosk (Alpha Cephei, 2022), a FOSS speech recognition toolkit for converting the audio from the video lecture to transcripts. This module takes a video lecture as an input and returns the associated transcript as an output. While this was an automated process, in its current version the library does not fully support the Greek language, which resulted in some errors. Consequently, the transcripts were inspected, and minor manual corrections were made. The data cleaning phase involved removing both redundant data and data that was not relevant for the analysis. After the requisite processing and clean-up, all the data that was extracted from the LMS was stored in a spreadsheet. We used the ML ecosystem that is built around the Python programming language and specifically the Pandas library for storing and preparing the data while the Scikit-Learn library was used for ML. The transcript of each video lecture that was extracted in the aforementioned step was also stored as text variable.

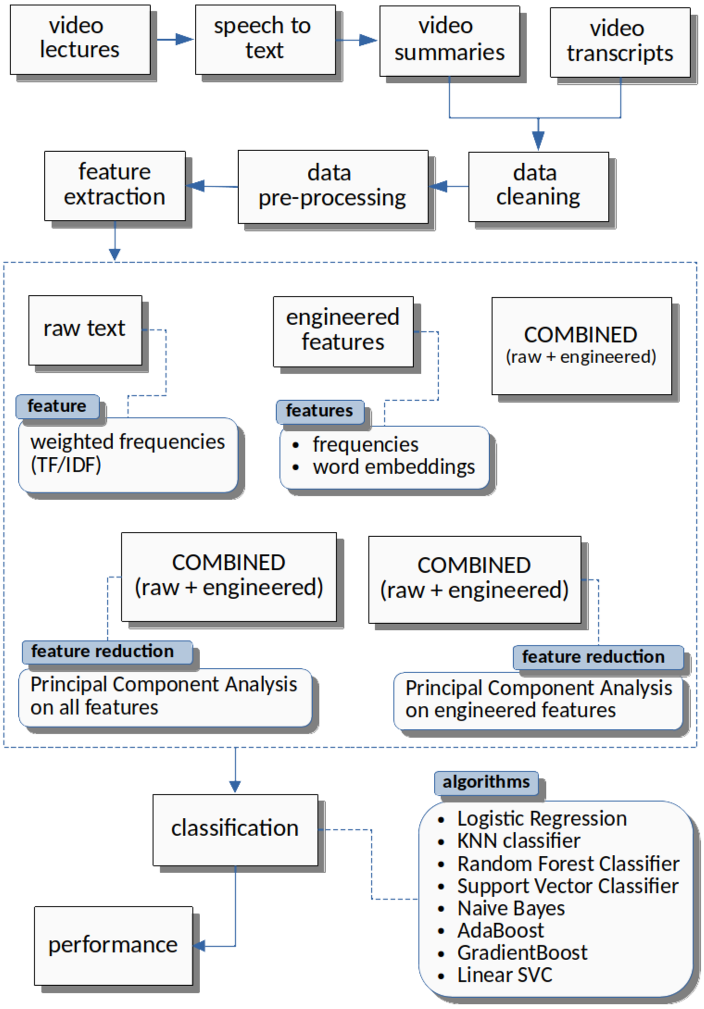

Figure 1. Outline of the workflow of the study

Student Performance on video lectures as a binary variable

First, the student performance on each video lecture was used to compute new aggregate variables. The median was then used to split each resulting variable into a binary one having two classes: low and high performance. The conversion of continuous numerical variables to binary ones for classification tasks is common in ML. For instance, Tomasevic et al. (2020) converted a 0-100 performance score into a binary one using the value of 40 as a cut-off point. Because of the distribution of the values in some performance variables, the median split did not always result in two classes with an approximately similar number of cases. As we wanted the two classes to be fairly balanced due to the small number of participants, we filtered the resulting classes manually, changing the threshold as needed to ensure that low and high-performance classes contained roughly equivalent student numbers.

Features

The study used five sets of features to train a series of ML algorithms for predicting performance. Principal Component Analysis (PCA) was also applied to the construction of two feature sets. PCA is a commonly used dimension reduction technique that transforms a set of correlated variables into a smaller set of uncorrelated ones. As a result, smaller data sets are easier to explore and make analysing data much easier and faster for ML algorithms without extraneous variables to process. Considering its simplicity, we used the raw text as a baseline model and proceeded by exploring the contributions made by adding various combinations of the other feature sets. The feature sets that were used in the study are described in the remainder of this section.

Raw text features: We considered the raw text (i.e., vectorized student summaries with weights) as a base feature set against which all other feature sets are compared. The raw text represents the simplest and most direct feature set that can be considered as a performance predictor because words are converted to vectors without any pre-processing whatsoever. In the case of raw text, each word used in the summary of a video lecture by a student is considered to be a feature of its own. Raw features have been extensively used by previous studies (Almatrafi et al., 2018; Wise et al., 2017).

More specifically, the first feature set consists of the raw text data that correspond to the student summaries. These raw student summaries were used directly as inputs for classification without any prior preprocessing (e.g., stopword removal or lemmatization). Vector representations of the summaries were created using the standard BoW approach, namely using token counts with normalized weights (TF-IDF with unigrams and bigrams). First, each summary was tokenized and a vocabulary was constructed from the words that occurred in the summary. Every word in the vocabulary was assigned a unique integer number. The frequency for each word in the vocabulary was calculated using the Term Frequency – Inverse Document Frequency method which reflects how much information is provided by words in the summary. The Term Frequency process is a measure of how frequently a given word appears in a summary and the Inverse Document Frequency will downscale words that appear frequently across all summaries. At the end of this process each summary was encoded as a sparse array with the normalized scores of the words to values between 0 and 1 with its length equal to the length of the vocabulary The Scikit – Learn library provides the TfidfVectorizer class that was used for parsing the student summaries and calculating the TF-IDF values.

Engineered text features: The second feature set consists of the engineered features that were extracted from students' summaries. In particular, we processed the raw text to derive a set of engineered features which provides a more comprehensive and nuanced picture of the data. The engineered features represent specific transformations or extractions of the raw text. Former studies have also examined engineered features such as word counts and sentence counts (Almatrafi et al., 2018; Dessì et al., 2019). Still, we introduced new features whose contributions have not been explored before: (a) grammatical information such as noun chunks (part of speech – POS) and (b) sematic information (distributed representations of words).

Regarding the former, the noun chunks do not carry any semantic information, that is they are grammar-related features. To the best of our knowledge, noun chunks have not been used as features in former studies for performance prediction. Regarding the latter, we have used cosine similarity for determining the association of each student summary with the corresponding video lecture. For each word in the student summaries of the video lectures an aggregate similarity measure was computed, which was then used as a feature for performance prediction. Unlike noun chunks, the similarity measure using word embeddings does capture the semantic meaning of words. The underlying concept is that words that have similar meanings tend to appear in similar contexts.

More specifically, to extract the features from the text, we created two groups of variables. As shown in Fig. 2, the first group was based on the frequencies of general linguistic features. In particular, the following variables were created: a) words, b) sentences, c) nouns, d) verbs and e) adjectives. It should be noted that while the first two variables represent general linguistic information, the last three ones convey Parts of Speech (POS) information.

The second group of variables is based on the semantic similarity between texts. For the purposes of this study, we have used pretrained word embeddings for the Greek language that are available through the spaCy library (Explosion, 2022). We have used spaCy's large language model for the Greek language (500K vectors) that was trained on a large corpus of words. In this model the meaning of each word is captured in 300 dimensions, i.e., each word is represented as a sequence of weights in a 300-dimensional vector. The spaCy library provides functions for finding the semantic similarity between

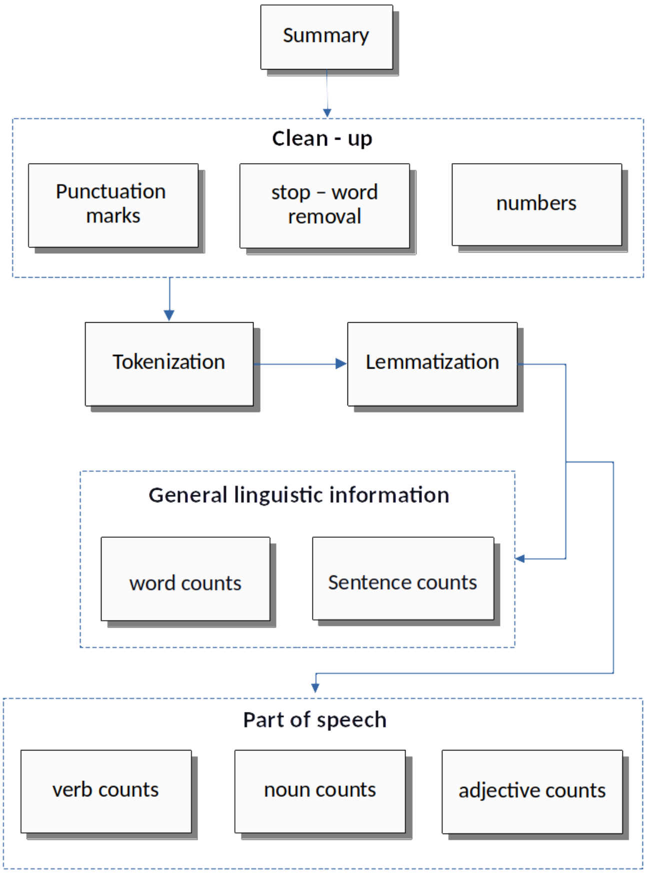

Figure 2. Data workflow for the first group of variables for feature engineering using the spaCy library

word vectors or any other spaCy objects. Word vectors can be compared to determine how closely related two words are in their meaning and usage. We used the cosine similarity which is a metric to determine the similarity between two words. Statistically, the cosine similarity metric measures the cosine of the angle between two n-dimensional vectors projected in a multi-dimensional space. The cosine similarity of two documents ranges from 0 to 1. The closer this value is to 1 the more similar two vectors are, i.e., have the same orientation. Conversely, the closer the value is to 0 the lower the similarity between the vectors. While primarily applicable to words, this similarity metric can also be used with larger units such as sentences. In the case of larger text chunks such as sentences, the semantic similarity is computed by averaging the token vectors in two spaCy objects respectively. While it is considered to be a satisfactory general-purpose measure, the averaging of token vectors is limited because it fails to capture the ordering of tokens (Explosion, 2022). In the present study, we compared two spaCy objects, the student summary of every video lecture and the video transcript of the respective lecture (see Fig. 3).

We created new variables in which different versions of the summary and the video lecture transcript were compared: (a) the raw text without any processing, (b) the preprocessed text, (c) the noun chunks from text, and (d) the preprocessed noun chunks from text (see Fig.4).

The preliminary evaluation of the results indicated three comparisons that yielded relatively high similarity values: a) pre-processed summary to pre-processed video transcript (ranging from 0.3 to 0.9), b) summary noun chunks to video transcript noun chunks (ranging from 0.3 to 0.9) and c) pre-processed summary to pre-processed video transcript noun chunks (ranging from 0.3 to 0.7). Figure 4 illustrates this process for each comparison. Consequently, these engineered features of the student summaries were first mined for each video lecture and then added to the Pandas data frame.

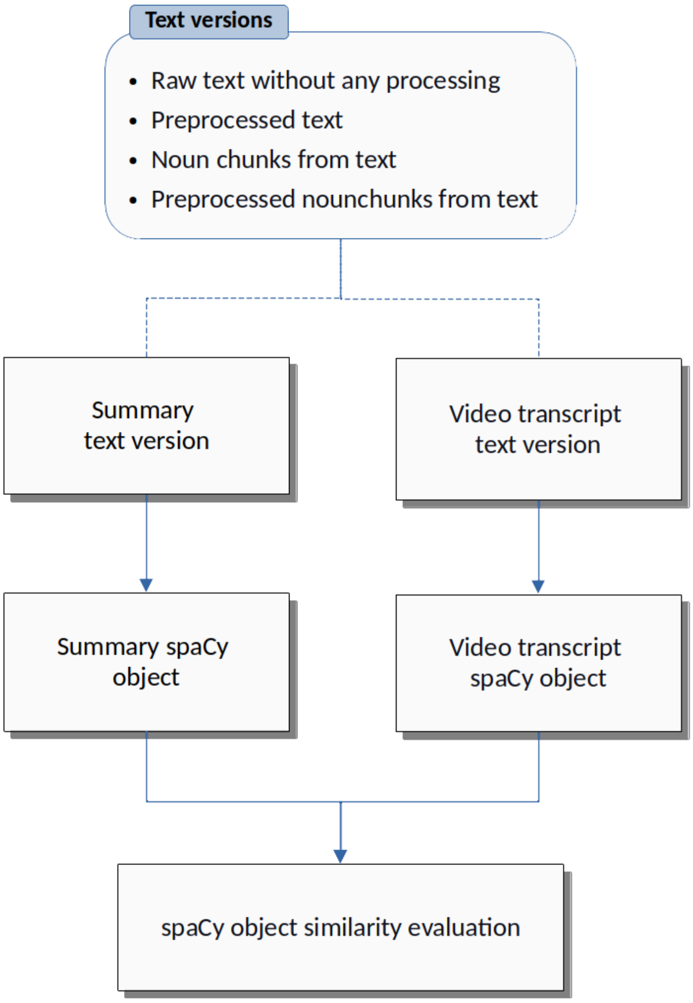
Text versions
Raw text without any processing
Preprocessed text
Noun chunks from text
Preprocessed nounchunks from text
Summary text version
Video transcript text version
Summary spaCy object
Video transcript spaCy object
spaCy object similarity evaluation

Figure 3. Comparison of summary and video transcript in different versions of the text

Combination of raw text features with engineered features: Considering that our preliminary analysis of the data indicated that the raw text features and engineered text features carry unique signals (Karasavvidis, Papadimas, & Ragazou, 2021), we have chosen to combine these different signals to determine their joint effect on performance prediction. Consequently, the third feature set consists of the combination of the raw data feature set with the engineered features set. The reasoning behind this category was to consider any possible signal overlap between the raw text features and the engineered features.

PCA-transformed combination of raw text features with engineered features: Considering the large number of features involved in NLP (e.g., each word in a 50-word summary is treated as a feature on its own), we followed standard feature reduction techniques, using a reduced feature set as predictors. Thus, the fourth feature set involved a dimensionality reduction of the whole combined feature set in step 3 above, i.e., of raw data features and engineered features. We aimed to explore whether a reduced set of features improves the performance prediction metrics.

Combination of raw text features with PCA – transformed engineered features: Lastly, the fifth feature set consists of the combination of raw data features and the feature reduced set of engineered features. The reasoning behind this choice was to examine if the initial set of engineered features could be replaced by a smaller one.

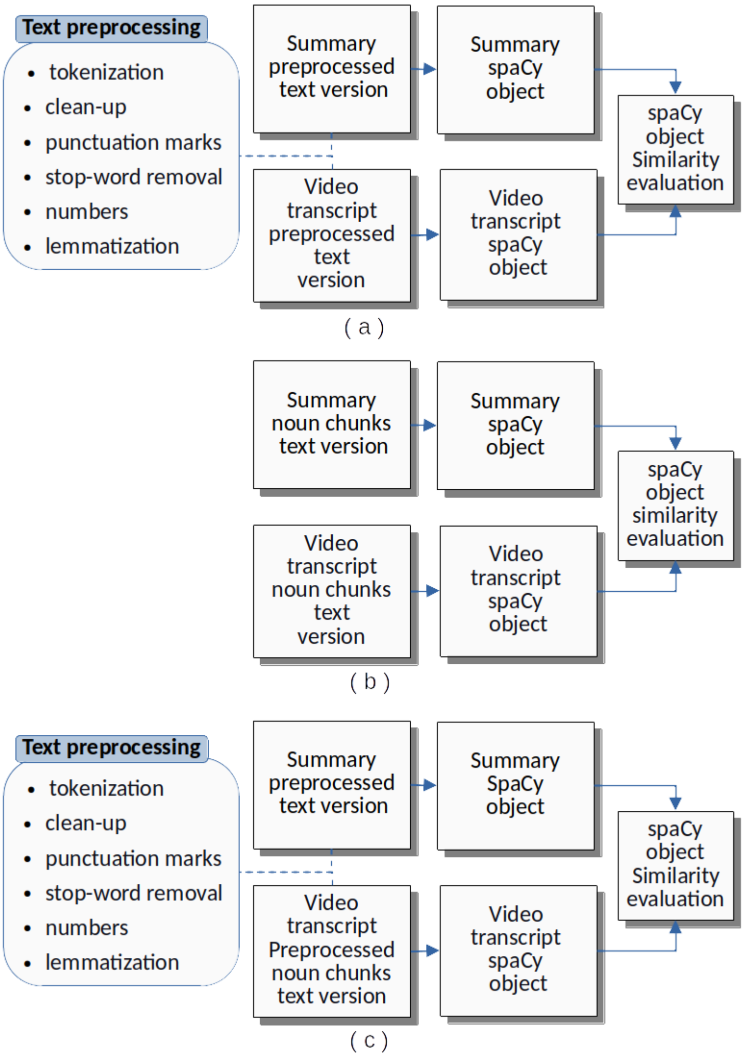
Text preprocessing
tokenization
clean-up
punctuation marks
stop-word removal
numbers
lemmatization
Summary preprocessed text version
Summary spaCy object
spaCy object Similarity evaluation
Video transcript preprocessed text version
Video transcript spaCy object
(a)
Summary noun chunks text version
Summary spaCy object
spaCy object similarity evaluation
Video transcript noun chunks text version
Video transcript spaCy object
(b)
Text preprocessing
tokenization
clean-up
punctuation marks
stop-word removal
numbers
lemmatization
Summary preprocessed text version
Summary SpaCy object
spaCy object Similarity evaluation
Video transcript Preprocessed noun chunks text version
Video transcript spaCy object
(c)

Figure 4. Data workflow of the comparisons for the second group of variables for feature engineering using the spaCy library. a) preprocessed summary to preprocessed video transcript, b) noun chunks from summary to noun chunks from video transcript, c) preprocessed summary to preprocessed noun chunks

Classification

Classification is a commonly used method to predict a data class for a given example of input data (Farid, Zhang, Rahman, Hossain & Strachan, 2014). Former studies have explored both binary (e.g., Dessì et al., 2019; Hasan et al., 2020) and multi-class classification (e.g., Yoon et al., 2021). In this case we used the aforementioned five feature sets to predict student performance into low (below median) and high (above median) classes. As this was an exploratory study, we have used the five feature sets to train a series of common ML classifiers: Logistic Regression, k-Nearest Neighbours (KNN), Random Forest, Support Vector Classifier (SVC), Naive Bayes (NV), AdaBoost, Gradient Boost, and Linear Support Vector Classifier (SVC). Following the standard convention, we used 90% of the data for training and validation and 10% of the data as a holdout test set.

Moreover, we used GridSearchCV to exhaustively search a large number of hyperparameters. This allowed the automatic fine-tuning of different algorithm hyperparameters to explore the potential of various feature set by algorithm combinations. Consequently, the accuracy metrics reported in this study need to be interpreted as the best obtainable ones. The performance of the classification algorithms was evaluated using the standard metrics: accuracy, precision, recall and F1-score.

Results

The comprehensive algorithm performance for accuracy is given in the Appendix (A). Table 2 shows the highest prediction obtained per video lecture regardless of algorithm. It is evident that all the feature sets achieve perfect prediction accuracy (1.0). Considering the frequency of occurrence of the highest accuracy prediction value (which is 1.0 in this case), we notice that certain feature sets yield perfect prediction in more than one video lectures. More specifically, (a) the raw data (R) features, (b) the combination of raw data features with engineered features (R+E) and (c) the feature-reduced combined feature set of raw data features with engineered features (FR(R+E)) resulted in the highest possible accuracy in most video lectures. The engineered features (E) yielded the highest prediction in two video lectures while the combination of raw data features with the feature reduced set of engineered features (R+FR(E)) only in one video lecture. It should be noted that the evaluation of the highest prediction accuracy in this table is only a first rough indicator of the information signal the feature sets carry and provides information about the best possible prediction.

Following others (Robinson et al., 2016; Tomasevic et al., 2020), we averaged the performance of the eight classifiers per video lecture. The average performance of all classifiers per feature set is presented in Table 3. As can be seen from the table, the raw data feature set (R) and the feature-reduced combined feature set (FR(R+E)) give the best average accuracy in three and two video lectures respectively. Interestingly, the engineered features (E) alone did not achieve a high average prediction in any video lecture.

On the other hand, it can be noted that in two video lectures, 2 and 6 respectively, the average prediction value is slightly above chance level. In the other four video lectures the average prediction ranges from 73% to 85%. The fourth video lecture with the use of raw data features yielded the highest average prediction accuracy (85%).

Table 2. Highest accuracy obtained per video (R – Raw, E – Engineered, FR() - feature reduction)

Video Lectures	R	E	R+E	FR(R+E)	R+FR(E)
v1	0.80	1.00	1.00	1.00	0.80
v2	1.00	0.57	1.00	0.40	0.80
v3	1.00	0.71	1.00	1.00	0.80
v4	1.00	0.80	1.00	0.80	1.00
v5	0.80	1.00	0.80	1.00	0.80
v6	0.80	0.80	0.60	0.80	0.80

Table 3. Average accuracy per video lecture (R – Raw, E – Engineered, FR - feature reduction)

Video Lectures	R	E	R+E	FR(R+E)	R+FR(E)
v1	0.58	0.70	0.75	**0.77**	0.71
v2	**0.63**[1]	0.32	0.48	0.26	0.40
v3	**0.70**	0.59	**0.70**	0.66	0.66
v4	**0.85**	0.56	0.60	0.66	0.71
v5	0.80	0.72	0.73	**0.83**	0.77
v6	0.43	0.55	0.53	0.60	**0.63**

[1] The highest values across all feature sets are given in bold

Table 4. Frequency distribution of the algorithms with the highest classification accuracy per video lecture

Algorithm	V1	V2	V3	V4	V5	V6	Sum	%
Gradient Boost	2	0	3	0	4	2	11	17.46
Random Forest Classifier	0	0	1	1	5	3	10	15.87
AdaBoost	2	1	1	2	3	1	10	15.87
KNN classifier	4	1	0	2	1	1	9	14.29
Linear SVC	0	1	0	1	3	3	8	12.70
Logistic Regression	0	0	1	1	3	1	6	9.52
Support Vector Classifier	1	3	0	0	2	0	6	9.52
Naive Bayes	0	0	0	3	2	0	5	7.94

Table 5. Average classification accuracy per algorithm over all video lectures per feature set

Algorithm	R[2]	E[2]	R+E	FR(R+E)[2]	R+FR(E)
Gradient Boost	**0.70**[1]	0.48	0.67	0.60	**0.70**
Random Forest Classifier	**0.77**	0.53	0.63	0.63	0.67
AdaBoost	**0.77**	0.59	0.67	0.60	0.73
KNN classifier	0.60	**0.63**	0.60	**0.63**	0.57
Linear SVC	0.56	**0.67**	0.63	**0.67**	0.60
Logistic Regression	0.57	0.63	0.63	**0.67**	0.60
Support Vector Classifier	0.63	0.60	0.63	0.60	**0.67**
Naive Bayes	**0.67**	0.43	0.57	-	-

[1] The highest values across all feature sets are given in bold

[2] R – Raw, E – Engineered, FR() - Feature Reduction

Table 4 presents a frequency distribution of the algorithms examined with respect to classification accuracy. As Table 4 illustrates, two of the three algorithms with the highest frequency of occurrence of higher average accuracy are boosting algorithms. Additionally, the Random Forest, the KNN, and the Linear SVC were also associated with relatively high frequencies of high average accuracy.

Table 5 presents the results of the average classification accuracy value across all video lectures for each feature set. As Table 5 shows, the raw data feature set obtained the highest average classification accuracy (77%). Moreover, the raw data feature set and the dimensionality-reduced combined feature set have the largest number of cases in which the classifiers yielded the highest average classification accuracy.

Discussion

The widespread use of E-learning systems in higher education has led to the capturing of the digital traces that students leave behind. These traces have been thoroughly studied in the course of the last decade, owning to the efforts of researchers in the fields of EDM and LA. Many LA studies have systematically examined the potential value of such data. A large number of studies have explored the predictive power of features such as student engagement and interaction with LMS including course logins (Agudo-Peregrina et al., 2014), clickstreams (Giannakos et al., 2015; Stöhr et al., 2019), and resource viewing (Agudo-Peregrina et al., 2014; Schumacher & Ifenthaler, 2018; Tomasevic et al., 2020). Former studies have indicated that such features could be used for many purposes such as predict final course performance, increase student retention, or identify at risk students (Huang et al., 2020; Kim et al., 2018; Tomasevic et al., 2020).

Even though students produce large volumes of text in E-learning systems, previous studies have not methodically explored text as a feature. The exception to this rule involves studies that have focused on forum discussions for predicting student performance (Conijn et al., 2017; Kim et al., 2016; Romero et al., 2013; Wang et al., 2015). However, most studies tend to extract contextual and behavioural features of forum posts, for example the number of posts or the number of words per post. The main drawback of such features is that they constitute general linguistic features that fail to take into consideration the semantic meaning of words.

The research described in this paper aimed to mitigate this problem by examining the potential of features that are extracted from student-produced texts for predicting student performance. In particular, we used NLP techniques and extracted five feature sets from short summaries that a cohort of undergraduates drafted after viewing a series of short video lectures. The two specific research questions that the study addressed are discussed below.

Text features that carry the most information signal for performance prediction

Regarding RQ1, the raw text feature set achieved higher average classification accuracy in two video lectures (3, 4), while in two other video lectures (1, 5) the combination with feature reduction resulted in higher classification accuracy. Additionally, the engineered text feature set achieved higher average accuracy classification in two video lectures (1, 5). The picture that emerges seems balanced as each set of text features carries an important signal for predicting student performance. A notable exception to this pattern concerns two video lectures (2, 6).

Firstly, it should be pointed out that the classification accuracy obtained is higher compared to what other studies, which employ linguistic features, report for binary classification tasks (e.g., Robinson et al., 2016). On the other hand, it is comparable with the accuracy metrics reported by other researchers (e.g., Almatrafi et al., 2018; Wise et al., 2016; Wu et al., 2020). However, it should be borne in mind that none of these studies was exclusively focused on predicting performance when learning from video lectures.

Secondly, the results indicate that the combination of raw and engineered text features is more efficient for predicting student performance compared to either feature set on its own. To some extent, this finding is to be expected considering that other studies also report that the combination of features results in higher classification accuracy. For example, Robinson et al. (2016) concluded that combining NLP feature with demographic ones resulted in higher classification accuracy. Similarly, Dessì et al. (2019) also found that the best classification results were obtained by combining frequency vectors (BoW) and semantic vectors (IBM Watson).

Thirdly, we meant to use the BoW approach as a baseline against which other feature combinations could be compared. Surprisingly enough, the raw text feature set turned out to yield the highest average accuracy in half of the video lectures. Other studies that have used linguistic features for training ML algorithms have also reached similar conclusions. For example, Dessì et al. (2019) intended to use TF-IDF as a baseline but reported that ultimately it performed quite well compared to the other feature sets, they had employed. Additionally, TF was also reported to perform well in the study by Almatrafi et al. (2018). Clearly, the signal that the raw text carry needs to be further systematically explored and replicated.

Finally, our findings indicate that classification accuracy was different across the six video lectures as a combination of feature sets and classifiers. Also, the eight classifiers obtained high accuracy prediction with the different text feature sets. This finding aligns with other studies that report that classification accuracy is dependent on the input features and classifiers used (Tomasevic et al., 2020). A possible reason for the variability of text features may lie in the topics of video lectures. Even though the video lecture series had a common theme, each lecture covered unique topics such as deep remixability or digital compositing (Manovich, 2013).

The performance of ML algorithms for classification

Following others, we evaluated various ML classifiers for predicting student performance based on five feature sets. The findings indicated that the Gradient Boost, AdaBoost and Random Forest classifiers achieved high average performance in three video lectures when trained on the raw text feature set. As discussed above, the potential of this feature set appears to be high and certainly worth further exploration. Regarding AdaBoost, we were only able to locate a single study that had used this algorithm with linguistic features (Almatrafi et al., 2018). The authors found that AdaBoost was the best classifier when using a combination of features that also included TF. While the specific accuracy value reported (0.87) by Almatrafi et al. (2018) is higher compared to the one obtained in the present study (0.77), our results are generally aligned with theirs considering that one of the two Boost classifiers tested in this study were amongst the top performing one's accuracy-wise.

With respect to the Random Forest algorithm, our results indicate an average classification accuracy of 0.77, which was one of the highest obtained by any of the classifiers used. It should be noted that this accuracy value is much higher than the one reported by Dessì et al. (2019), though it should be acknowledged that their study involved a multi-class classification task. Almatrafi et al. (2018) also reported that the Random Forest classifier was second in terms of accuracy in their binary classification task (i.e., predicting when a forum poster requires an urgent response or not).

Moreover, KNN and Linear SVC performed moderately well with two feature sets, the engineered feature set and the combined feature set with reduced dimensionality. Considering the lack of studies employing these classifiers, it is difficult to contextualize our findings.

As far as Logistic Regression is concerned, our findings suggest that the combination of Logistic Regression with the combined feature set whose dimensionality had been reduced achieved an average accuracy of .67 which is higher than what Robinson et al. (2016) reported in their study. However, the performance of Logistic Regression with this particular feature set was lower compared to what is reported by other studies that employ linguistic features. For instance, using Logistic Regression both Wise et al. (2016) and Almatrafi et al. (2018) obtained an accuracy of 0.80 for binary classification tasks.

The average classification performance of SVM and NB was also moderate (0.67) using two feature sets as inputs, raw feature set (NB) and combined feature set (SVM) on which feature reduction has been applied to the engineered features only. The performance of these classifiers is comparable to the findings of other studies that use linguistic features as input for ML algorithms (e.g., Almatrafi et al., 2018).

Study implications and future work

Overall, the findings from this exploratory study suggest that the extraction of features from student-generated texts is very promising. In the long run the proposed approach is expected to have two major real-world applications. First, the aspiration is that it will enable real-time inferences of student understanding based on written summaries and responses to open-ended questions in general. Second, it will pave the way for more personalized feedback, one that is not generic or pre-packaged but based on dynamic evaluations of student understanding and performance.

The approach introduced in this work is not directly applicable to face-to-face settings. Capturing student written responses and analysing them in real time to determine how well they correspond to an ongoing teacher lecture is challenging in some respects. While it is technically feasible to apply some of the components of the proposed method to live lectures (e.g., use cell phones to scan hand-written responses to open questions or use voice recognition applications to convert the lecture audio to text), vectorizing texts and running certain computationally heavy machine learning models in real time is impractical. However, the proposed approach is perfectly applicable to hybrid or fully online learning settings. In such cases, three main applications can be identified: system-based, instructor-based, and learner-based.

With respect to system-based applications, the study findings could inform learning analytics and the design of various forms of feedback and remedial instruction. More specifically, students could first watch a video lecture and then summarize the main points or respond to a series of questions. Then, the system could either directly use these student-generated texts as features or distil features from these to determine their similarity to the lecture transcripts. Next, a combination of such raw and engineered features could be used by the system to gauge student understanding and predict performance. Based on the outcomes of such predictions, the system could further provide targeted feedback and support (e.g., worked-out examples, summaries of certain concepts, prompts to revisit specific portions of the video lectures in which concepts are introduced or applied, direct students to specific materials to review, recommend certain practice tasks etc.). The idea is to assign students to extra course work that could help achieve mastery.

Regarding instructor-based applications, the outcomes of this study are directly interpretable and actionable, as they clearly illustrate the extent to which students have understood the material. As opposed to standard quantitative measures (e.g., quizzes) that are necessarily static, the feature sets introduced in this study offer educators a broader range of measures that are more dynamic and nuanced in nature. Based on such dynamic measures, the instructors can either design remedial interventions using appropriate feedback strategies or take any measures that are deemed appropriate for improving student understanding. For instance, educators could devise additional tasks so that the students could practice mastering the requisite concepts.

Finally, the study findings can be of potential interest to the learners themselves. More specifically, the results could provide students with information on concepts or topics that are not well understood or only partially understood. Based on this information, the students could then engage in autonomous, self-regulated study and review, such as revisiting specific portions of the video lectures, reviewing certain materials, or practicing further on an additional set of tasks,

What is important to note is that, ultimately, both the measurement of students' conceptual understanding and the subsequent feedback that could be provided can be both qualitative and dynamic in nature. Regarding the former, the evaluation of understanding will not be based on performance metrics alone (e.g., 60% score on a quiz), namely it will be evaluated using student-generated texts. For instance, Mangaroska and Giannakos (2018) stressed the need to use complimentary metrics to the standard quantitative features. Considering that this requirement is met through the utilization of student-authored texts, in the long run this approach might offer a more qualitative outlook of student understanding. Tapping on the potential of student generated responses (such as summaries or responses to open ended questions) opens up a new way of determining student comprehension levels.

Regarding the latter, the evaluation of understanding will be more dynamic, flexible, and tailored to individual students and their needs. Take for example the study by (Matcha, Uzir, Gasevic, & Pardo, 2020) in which the instructors had prepared in advance a set of comments for students exhibiting different engagement levels with the video lectures, namely students who only glanced through the video lecture, students who watched a small portion, students who viewed the whole video, and students who viewed the video multiple times. The underlying assumption behind their approach is that different student engagement levels require different levels of feedback. A potential limitation of such approaches of pre-packaged feedback is that it might correspond to e.g., student engagement levels but not necessarily to students' conceptual understanding. For example, students might skip portions of a video lecture because they are familiar with the concepts or because they expect the concepts to make more sense in the next sections. The approach introduced in this work could easily mitigate such limitations through the provision of automated feedback that is personalized. For instance, given a quiz question that students fail to answer correctly, NLP could be used to extract an answer from the complete text corpus used in the course (i.e., video lecture transcripts, lecture slides, course notes, and course readings). On the other hand, if the approach introduced in this work turns out to be consistently effective in predicting student performance of student-authored texts, then the provision of feedback that is automated and real-time might be a possibility. To put this in perspective, let us consider former studies that have employed both non-linguistic features (Romero et al., 2013; Rossi & Gnawali, 2014; Wen et al., 2014) and linguistic ones (Almatrafi et al., 2018; Dessì et al., 2019; Robinson et al., 2016; Wise et al., 2016). In terms of feedback, the main limitation of such studies is that sufficient data needs to be collected over a span of several weeks or even months before the instructors can be informed about whether a particular student is facing difficulties, is struggling with the course content or is likely to drop out. As opposed to such extended time frames, a refined version of the approach outlined in this work could provide almost real-time information of students' understanding of video lectures and facilitate the provision of usable feedback, which is genuinely based on students' conceptual understanding rather than pre-packaged.

Limitations

Despite the novel approach concerning text-based feature sets adopted by the present study and the promise of student-produced texts for predicting learning from video lectures, the findings should be interpreted with caution. The first main limitation concerns the number of participants. Although several other ML studies have used comparable numbers of participants (Abu Zohair, 2019; Wu et al., 2020; Yoon, Lee & Jo, 2021), we plan to replicate and extend our findings using a larger dataset. A second limitation is that the study was not balanced in terms of gender. Because the research was conducted in a preschool childhood education department, the female to male ratio of the student population is necessarily reflected in the sample participants, resulting in a very high percentage of female students. Replication of the findings with a more gender- balanced sample is desirable. A third limitation is that we focused exclusively on text-based features, ignoring a variety of LMS behavioural indicators (Mangaroska, Sharma, Gašević & Giannakos, 2020) or learner attributes (i.e., mental effort, self-efficacy, motivation, flow) that could be used as indicators. For example, Schumacher and Ifenthaler (2018) found that students' motivational dispositions played a crucial role in providing a personalized model that could support learning and motivation. Our future plans involve the examination of learner attributes in combination with text-based features for predicting student performance.

Fourth, the method proposes in this work requires extensive human intervention for data processing, running ML models, and interpreting the results. Currently, there is considerable manual labour and intervention involved because the level of automation is low. However, in the long run – if such feature sets turn out to be consistently conducive to student performance prediction – they can be standardized and streamlined (e.g., develop a series of scripts to automate the process, design and develop a corresponding module for integration in LMS systems).

Last, in its current form the proposed approach presupposes video lecture transcripts. If no video lectures are available, then in principle other course materials could be used (e.g., lecture slides, course readings, other literature papers). However, we have not yet tested if a wider range of course-related texts could be as effective for deriving features and determining the similarity with student summaries or responses to open-ended questions. While it seems plausible to assume that the feature sets extracted from e.g., course readings could be equivalent to video lecture transcripts, this needs to be systematically explored.

Overall, our work needs to be seen as exploratory given that it covers relatively new ground, is far from conclusive, and non-standardized. As this is one of the first studies to tackle performance prediction using student-generated texts, more work is clearly needed to verify the real contribution each set of text features makes performance-wise. Replications in similar contexts and extensions to other academic subjects and topics are required to determine the actual added value of this approach.

Conclusion

To date, we are not aware of studies using student-generated texts as features for either predicting student performance or gauging student understanding in online learning settings. This paper introduced a novel approach for predicting student performance after viewing short video lectures. Our approach consisted in using five text-based feature sets to train eight ML classifiers. The findings indicated that the raw text feature set achieved higher average classification accuracy in two video lectures, while the combined feature set whose dimensionality had been reduced resulted in higher classification accuracy in two other video lectures. In terms of algorithms, the findings indicated that the Gradient Boost, AdaBoost and Random Forest classifiers achieved high average performance in half of the video lectures. Overall, our findings suggest the potential of using unstructured text data produced by students for predicting performance when learning from video lectures. The present work paves the way for a new methodical approach to the use of text-based features for predicting student performance in e-learning environments.

Acknowledgements

This research is co-financed by Greece and the European Union (European Social Fund- ESF) through the Operational Programme "Human Resources Development, Education and Lifelong Learning 2014-2020" in the context of the project “Planning, Development and Deployment of an Intelligent Feedback System Using Supervised and Unsupervised Machine Learning Methods” (MIS 5048955).

The authors would like to express their sincere gratitude to all the students who responded to our call for participation in the study amidst the harsh COVID19 pandemic conditions. Finally, the authors would also like to thank the editor and two anonymous reviewers whose critical comments and detailed suggestions substantially helped improving the original manuscript.

References

Abu Zohair, L. M. (2019). Prediction of Student’s performance by modelling small dataset size. *International Journal of Educational Technology in Higher Education*, *16*(1), 27. https://doi.org/10.1186/s41239-019-0160-3.

Agudo-Peregrina, Á. F., Hernández-García, Á., & Pascual-Miguel, F. J. (2014). Behavioral intention, use behavior and the acceptance of electronic learning systems: Differences between higher education and lifelong learning. *Computers in Human Behavior*, *34*, 301–314. https://doi.org/10.1016/j.chb.2013.10.035.

Almatrafi, O., Johri, A., & Rangwala, H. (2018). Needle in a haystack: Identifying learner posts that require urgent response in MOOC discussion forums. *Computers & Education*, *118*, 1–9. https://doi.org/10.1016/j.compedu.2017.11.002.

Alpha Cephei (2022). [Computer software]. Retrieved May 22, 2022, from https://alphacephei.com/vosk.

Blikstein, P., Worsley, M., Piech, C., Sahami, M., Cooper, S., & Koller, D. (2014). Programming pluralism: Using learning analytics to detect patterns in the learning of computer programming. *Journal of the Learning Sciences*, *23*(4), 561–599. https://doi.org/10.1080/10508406.2014.954750.

Chen, B., Chang, Y.-H., Ouyang, F., & Zhou, W. (2018). Fostering student engagement in online discussion through social learning analytics. *The Internet and Higher Education*, *37*, 21–30. https://doi.org/10.1016/j.iheduc.2017.12.002.

Conijn, R., Snijders, C., Kleingeld, A., & Matzat, U. (2017). Predicting student performance from LMS data: A comparison of 17 blended courses using Moodle LMS. *IEEE Transactions on Learning Technologies*, *10*(1), 17–29. https://doi.org/10.1109/TLT.2016.2616312.

Cui, Y., & Wise, A. F. (2015). Identifying Content-Related Threads in MOOC Discussion Forums. *Proceedings of the Second (2015) ACM Conference on Learning @ Scale*, 299–303. New York, NY, USA: ACM. https://doi.org/10.1145/2724660.2728679.

Dessì, D., Fenu, G., Marras, M., & Reforgiato Recupero, D. (2019). Bridging learning analytics and Cognitive Computing for Big Data classification in micro-learning video collections. *Computers in Human Behavior*, *92*, 468–477. https://doi.org/10.1016/j.chb.2018.03.004.

Farid, D. Md., Zhang, L., Rahman, C. M., Hossain, M. A., & Strachan, R. (2014). Hybrid decision tree and naïve Bayes classifiers for multi-class classification tasks. *Expert Systems with Applications*, *41*(4), 1937–1946. https://doi.org/10.1016/j.eswa.2013.08.089.

Ferguson, R. (2012). Learning analytics: drivers, developments and challenges. *International Journal of Technology Enhanced Learning*, *4*(5/6), 304. https://doi.org/10.1504/IJTEL.2012.051816.

Gašević, D., Dawson, S., Rogers, T., & Gasevic, D. (2016). Learning analytics should not promote one size fits all: The effects of instructional conditions in predicting academic success. *The Internet and Higher Education*, *28*, 68–84. https://doi.org/10.1016/j.iheduc.2015.10.002.

Giannakos, M. N., Chorianopoulos, K., & Chrisochoides, N. (2014). Collecting and making sense of video learning analytics. *In 2014 IEEE Frontiers in Education Conference (FIE) Proceedings* (pp. 1-7). IEEE.

Hasan, R., Palaniappan, S., Mahmood, S., Abbas, A., Sarker, K. U., & Sattar, M. U. (2020). Predicting student performance in higher educational institutions using video learning analytics and data mining techniques. *Applied Sciences*, *10*(11), 3894. https://doi.org/10.3390/app10113894.

Hayati, H., Chanaa, A., Khalidi Idrissi, M., & Bennani, S. (2019). Doc2Vec & Naïve Bayes: Learners' cognitive presence assessment through asynchronous online discussion TQ transcripts. *International Journal of Emerging Technologies in Learning*, *14*(08), 70. https://doi.org/10.3991/ijet.v14i08.9964.

Hernández-García, Á., & Conde-González, M. Á. (2016). Bridging the gap between LMS and Social Network Learning Analytics in online learning. *Journal of Information Technology Research*, *9*(4), 1–15. https://doi.org/10.4018/JITR.2016100101.

Huang, A. Y. Q., Lu, O. H. T., Huang, J. C. H., Yin, C. J., & Yang, S. J. H. (2020). Predicting students' academic performance by using educational big data and learning analytics: evaluation of classification methods and learning logs. *Interactive Learning Environments*, *28*(2), 206–230. https://doi.org/10.1080/10494820.2019.1636086.

Ifenthaler, D., & Widanapathirana, C. (2014). Development and Validation of a Learning Analytics Framework: Two Case Studies Using Support Vector Machines. *Technology, Knowledge and Learning*, *19*(1–2), 221–240. https://doi.org/10.1007/s10758-014-9226-4.

Karasavvidis, I., Papadimas, C., & Ragazou, V. (2021). A comparison of two text-based feature sets for predicting student performance: An initial exploration. *14th Annual International Conference of Education, Research and Innovation* (pp. 4806-4814). Online Conference: IATED.

Kim, D., Park, Y., Yoon, M., & Jo, I.-H. (2016). Toward evidence-based learning analytics: Using proxy variables to improve asynchronous online discussion environments. *The Internet and Higher Education*, *30*, 30–43. https://doi.org/10.1016/j.iheduc.2016.03.002.

Kim, D., Yoon, M., Jo, I.-H., & Branch, R. M. (2018). Learning analytics to support self-regulated learning in asynchronous online courses: A case study at a women's university in South Korea. *Computers & Education*, *127*, 233–251. https://doi.org/10.1016/j.compedu.2018.08.023.

Mangaroska, K., Sharma, K., Gašević, D., & Giannakos, M. (2020). Multimodal Learning Analytics to inform learning design: Lessons learned from computing education. *Journal of Learning Analytics*, *7*(3), 79-97. https://doi.org/10.18608/jla.2020.73.7.

Manovich, L. (2013). *Software Takes Command* (Vol. 5). A&C Black.

Matcha, W., Uzir, N. A., Gasevic, D., & Pardo, A. (2020). A systematic review of empirical studies on Learning Analytics dashboards: A self-regulated learning perspective. *IEEE Transactions on Learning Technologies*, *13*(2), 226–245. https://doi.org/10.1109/TLT.2019.2916802.

Mayer, R. E., Fiorella, L., & Stull, A. (2020). Five ways to increase the effectiveness of instructional video. *Educational Technology Research and Development*, *68*(3), 837-852. https://doi.org/10.1007/s11423-020-09749-6.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). *Efficient Estimation of Word Representations in Vector Space*. Retrieved from http://arxiv.org/abs/1301.3781.

Papamitsiou, Z., & Economides, A. (2014). Learning analytics and educational data mining in practice: A systematic literature review of empirical evidence. *Educational Technology & Society*, *17*(4), 49–64.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., & Thirion, B. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Rienties, B., Cross, S., & Zdrahal, Z. (2017). Implementing a Learning Analytics intervention and evaluation framework: What works? In *Big Data and Learning Analytics in Higher Education* (pp. 147–166). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-06520-5_10.

Robinson, C., Yeomans, M., Reich, J., Hulleman, C., & Gehlbach, H. (2016). Forecasting student achievement in MOOCs with natural language processing. *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge - LAK '16*, 383–387. New York, New York, USA: ACM Press. https://doi.org/10.1145/2883851.2883932.

Romero, C., López, M.-I., Luna, J.-M., & Ventura, S. (2013). Predicting students' final performance from participation in on-line discussion forums. *Computers & Education*, *68*, 458–472. https://doi.org/10.1016/j.compedu.2013.06.009.

Romero, C., & Ventura, S. (2010). Educational data mining: A review of the state of the art. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 40(6), 601–618. https://doi.org/10.1109/TSMCC.2010.2053532.

Rossi, L. A., & Gnawali, O. (2014). Language independent analysis and classification of discussion threads in Coursera MOOC forums. In *Proceedings of the 2014 IEEE 15th International Conference on Information Reuse and Integration* (pp. 654–661). Redwood City, CA, USA: IEEE. https://doi.org/10.1109/IRI.2014.7051952.

Schumacher, C., & Ifenthaler, D. (2018). Features students really expect from learning analytics. *Computers in Human Behavior*, *78*, 397–407. https://doi.org/10.1016/j.chb.2017.06.030.

Explosion (2022). *spaCy* (v3.0) [Computer software]. Retrieved May 22, 2022, from https://spacy.io.

Stöhr, C., Stathakarou, N., Mueller, F., Nifakos, S., & McGrath, C. (2019). Videos as learning objects in MOOCs: A study of specialist and non-specialist participants' video activity in MOOCs. *British Journal of Educational Technology*, *50*(1), 166–176. https://doi.org/10.1111/bjet.12623.

Tempelaar, D., Rienties, B., Mittelmeier, J., & Nguyen, Q. (2018). Student profiling in a dispositional learning analytics application using formative assessment. *Computers in Human Behavior*, *78*, 408–420. https://doi.org/10.1016/j.chb.2017.08.010.

Tomasevic, N., Gvozdenovic, N., & Vranes, S. (2020). An overview and comparison of supervised data mining techniques for student exam performance prediction. *Computers & Education*, *143*, 103676. https://doi.org/10.1016/j.compedu.2019.103676.

Wang, Y., Liu, S., Afzal, N., Rastegar-Mojarad, M., Wang, L., Shen, F., … Liu, H. (2018). A comparison of word embeddings for the biomedical natural language processing. *Journal of Biomedical Informatics*, *87*, 12–20. https://doi.org/10.1016/j.jbi.2018.09.008.

Wang, Y.-C., Kraut, R. E., & Levine, J. M. (2015). Eliciting and receiving online support: Using computer-aided content analysis to examine the dynamics of online social support. *Journal of Medical Internet Research*, *17*(4), e99. https://doi.org/10.2196/jmir.3558.

Wen, M., Yang, D., & Rose, C. (2014). Linguistic reflections of student engagement in Massive Open Online Courses. *Proceedings of the International AAAI Conference on Web and Social Media*, *8*(1), 525–534. Retrieved from https://ojs.aaai.org/index.php/ICWSM/article/view/14512.

Wise, A. F., Cui, Y., Jin, W., & Vytasek, J. (2017). Mining for gold: Identifying content-related MOOC discussion threads across domains through linguistic modeling. *The Internet and Higher Education*, *32*, 11-28. https://doi.org/10.1016/j.iheduc.2016.08.001.

Wise, A. F., Cui, Y., & Vytasek, J. (2016). Bringing order to chaos in MOOC discussion forums with content-related thread identification. *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge - LAK '16* (pp. 188–197). New York, USA: ACM Press. https://doi.org/10.1145/2883851.2883916.

Wu, J.-Y., Hsiao, Y.-C., & Nian, M.-W. (2020). Using supervised machine learning on large-scale online forums to classify course-related Facebook messages in predicting learning achievement within the personal learning environment. *Interactive Learning Environments*, *28*(1), 65–80. https://doi.org/10.1080/10494820.2018.1515085.

Yang, D., Wen, M., Howley, I., Kraut, R., & Rose, C. (2015). Exploring the effect of confusion in discussion forums of Massive Open Online Courses. *Proceedings of the Second (2015) ACM Conference on Learning @ Scale* (pp. 121–130). New York, USA: ACM. https://doi.org/10.1145/2724660.2724677.

Yoon, M., Lee, J., & Jo, I.-H. (2021). Video learning analytics: Investigating behavioral patterns and learner clusters in video-based online learning. *The Internet and Higher Education*, *50*, 100806. https://doi.org/10.1016/j.iheduc.2021.100806.

Zhang, H., & Litman, D. (2018). Co-Attention Based Neural Network for Source-Dependent Essay Scoring. *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications* (pp. 399–409). Stroudsburg, PA, USA: Association for Computational Linguistics. https://doi.org/10.18653/v1/W18-0549.

Appendix A. Algorithm performance metrics

Table A1. Performance metrics for Video 1 per feature set (R – Raw, E – Engineered, FR() - feature reduction)

Algorithms	R				E				R+E				FR(R+E)				R+FR(E)			
	A	P	R	F	A	P	R	F	A	P	R	F	A	P	R	F	A	P	R	F
Logistic Regression	0.40	0.00	0.00	0.00	0.80	1.00	0.67	0.80	0.80	1.00	0.67	0.80	0.80	1.00	0.67	0.80	0.60	0.67	0.67	0.67
KNN classifier	0.40	0.00	0.00	0.00	**1.00**	1.00	1.00	1.00	**1.00**	1.00	1.00	1.00	**1.00**	1.00	1.00	1.00	**0.80**	0.75	1.00	0.86
Random Forest	0.60	1.00	0.34	0.50	0.60	1.00	0.34	0.50	0.80	1.00	0.67	0.80	0.60	1.00	0.33	0.50	0.60	1.00	0.33	0.50
Support Vector Classifier	0.60	0.60	1.00	0.75	0.80	1.00	0.67	0.80	0.80	1.00	0.67	0.80	0.80	1.00	0.67	0.80	**0.80**	0.75	1.00	0.86
Naive Bayes	0.60	1.00	0.34	0.50	0.40	0.00	0.00	0.00	0.40	0.50	0.34	0.40								
AdaBoost	**0.80**[1]	1.00	0.67	0.80	0.57	0.67	0.50	0.57	0.60	1.00	0.33	0.50	0.60	1.00	0.33	0.50	**0.80**	1.00	0.67	0.80
Gradient Boost	**0.80**	1.00	0.67	0.80	0.60	1.00	0.33	0.50	0.80	1.00	0.67	0.80	0.80	0.80	1.00	0.67	**0.80**	1.00	0.67	0.80
Linear SVC	0.40	0.00	0.00	0.00	0.80	1.00	0.67	0.80	0.80	1.00	0.67	0.80	0.80	1.00	0.67	0.80	0.60	0.67	0.67	0.67

[1].The highest values across all feature sets are given in bold, A: Accuracy, P: Precision, R: Recall, F: F1 score

Table A2. Performance metrics for Video 2 per feature set (R – Raw, E – Engineered, FR() - feature reduction)

Algorithms	R				E				R+E				FR(R+E)				R+FR(E)			
	A	P	R	F	A	P	R	F	A	P	R	F	A	P	R	F	A	P	R	F
Logistic Regression	0.20	0.00	0.00	0.00	0.40	1.00	0.25	0.40	0.20	0.00	0.00	0.00	0.20	0.00	0.00	0.00	0.20	0.00	0.00	0.00
KNN classifier	**1.00**[1]	1.00	1.00	1.00	0.20	0.00	0.00	0.00	0.20	0.00	0.00	0.00	0.20	0.00	0.00	0.00	0.20	0.00	0.00	0.00
Random Forest	0.80	1.00	0.75	0.86	0.20	0.00	0.00	0.00	0.40	1.00	0.25	0.40	0.20	0.00	0.00	0.00	0.20	0.00	0.00	0.00
Support Vector Classifier	0.80	0.80	1.00	0.89	0.40	1.00	0.25	0.40	**1.00**	1.00	1.00	1.00	**0.40**	1.00	0.25	0.40	**0.80**	0.80	1.00	0.89
Naive Bayes	0.60	1.00	0.50	0.67	0.20	0.00	0.00	0.00	0.20	0.00	0.00	0.00								
AdaBoost	0.80	1.00	0.75	0.86	**0.57**	1.00	0.40	0.57	0.80	1.00	0.75	0.86	0.20	0.00	0.00	0.00	0.60	1.00	0.50	0.67
Gradient Boost	0.60	1.00	0.50	0.67	0.20	0.00	0.00	0.00	0.60	1.00	0.50	0.67	0.20	0.00	0.00	0.00	0.40	1.00	0.25	0.40
Linear SVC	0.20	0.00	0.00	0.00	0.40	1.00	0.25	0.40	0.40	1.00	0.25	0.40	**0.40**	1.00	0.25	0.40	0.40	1.00	0.25	0.40

[1].The highest values across all feature sets are given in bold, A: Accuracy, P: Precision, R: Recall, F: F1 score

Table A3. Performance metrics for Video 3 per feature set (R – Raw, E – Engineered, FR() - feature reduction)

Algorithms	R				E				R+E				FR(R+E)				R+FR(E)			
	A	P	R	F	A	P	R	F	A	P	R	F	A	P	R	F	A	P	R	F
Logistic Regression	0.60	1.00	0.33	0.50	0.60	1.00	0.33	0.50	0.80	1.00	0.67	0.80	**1.00**	1.00	1.00	1.00	0.60	1.00	0.33	0.50
KNN classifier	0.60	1.00	0.33	0.50	0.60	1.00	0.33	0.50	0.60	1.00	0.33	0.50	0.60	1.00	0.33	0.50	0.60	1.00	0.33	0.50
Random Forest	0.80	1.00	0.67	0.80	0.60	1.00	0.33	0.50	0.60	1.00	0.33	0.50	0.60	1.00	0.33	0.50	**0.80**	1.00	0.67	0.80
Support Vector Classifier	0.60	1.00	0.33	0.50	0.60	1.00	0.33	0.50	0.60	1.00	0.33	0.50	0.60	1.00	0.33	0.50	0.60	0.60	1.00	0.75
Naive Bayes	0.60	1.00	0.33	0.50	0.40	0.00	0.00	0.00	0.60	1.00	0.33	0.50								
AdaBoost	0.60	1.00	0.33	0.50	**0.71**	1.00	0.33	0.50	0.80	1.00	0.67	0.80	0.60	1.00	0.33	0.50	0.60	1.00	0.33	0.50
Gradient Boost	**1.00**[1]	1.00	1.00	1.00	0.6	1.00	0.33	0.50	**1.00**	1.00	1.00	1.00	0.60	1.00	0.33	0.50	**0.80**	1.00	0.67	0.80
Linear SVC	0.6	1.00	0.33	0.50	0.60	1.00	0.33	0.50	0.60	1.00	0.33	0.50	0.60	1.00	0.33	0.50	0.60	1.00	0.33	0.50

[1] The highest values across all feature sets are given in bold, A: Accuracy, P: Precision, R: Recall, F: F1 score

Table A4. Performance metrics for Video 4 per feature set (R – Raw, E – Engineered, FR() - feature reduction)

Algorithms	R				E				R+E				FR(R+E)				R+FR(E)			
	A	P	R	F	A	P	R	F	A	P	R	F	A	P	R	F	A	P	R	F
Logistic Regression	**1.00**[1]	1.00	1.00	1.00	0.60	1.00	0.50	0.67	0.60	1.00	0.50	0.67	0.60	1.00	0.50	0.67	0.80	1.00	0.75	0.86
KNN classifier	0.60	1.00	0.50	0.67	**0.80**	1.00	0.75	0.86	0.60	1.00	0.50	0.67	**0.80**	1.00	0.75	0.86	0.60	1.00	0.50	0.67
Random Forest	0.80	1.00	0.75	0.86	0.20	0.00	0.00	0.00	0.60	1.00	0.50	0.67	**0.80**	1.00	0.75	0.86	0.80	1.00	0.75	0.86
Support Vector Classifier	0.80	0.80	1.00	0.89	0.60	1.00	0.50	0.67	0.60	1.00	0.50	0.67	0.60	1.00	0.50	0.67	0.60	1.00	0.50	0.67
Naive Bayes	**1.00**	1.00	1.00	1.00	**0.80**	0.80	1.00	0.89	**1.00**	1.00	1.00	1.00								
AdaBoost	**1.00**	1.00	1.00	1.00	0.71	0.80	0.80	0.80	0.60	1.00	0.50	0.67	0.60	1.00	0.50	0.67	**1.00**	1.00	1.00	1.00
Gradient Boost	0.60	1.00	0.50	0.67	0.20	0.00	0.00	0.00	0.20	0.00	0.00	0.00	0.60	1.00	0.50	0.67	0.60	1.00	0.50	0.67
Linear SVC	**1.00**	1.00	1.00	1.00	0.60	1.00	0.50	0.67	0.60	1.00	0.50	0.67	0.60	1.00	0.50	0.67	0.60	1.00	0.50	0.67

[1] The highest values across all feature sets are given in bold, A: Accuracy, P: Precision, R: Recall, F: F1 score

Table A5. Performance metrics for Video 5 per feature set (R – Raw, E – Engineered, FR() - feature reduction)

Algorithms	R				E				R+E				FR(R+E)				R+FR(E)			
	A	P	R	F	A	P	R	F	A	P	R	F	A	P	R	F	A	P	R	F
Logistic Regression	**0.80**[1]	0.75	1.00	0.86	0.80	0.75	1.00	0.86	**0.80**	0.75	1.00	0.86	0.80	0.75	1.00	0.86	**0.80**	0.75	1.00	0.86
KNN classifier	**0.80**	0.75	1.00	0.86	0.60	0.67	0.67	0.67	0.60	0.67	0.67	0.67	0.60	0.67	0.67	0.67	0.60	0.67	0.67	0.67
Random Forest	**0.80**	0.75	1.00	0.86	**1.00**	1.00	1.00	1.00	**0.80**	0.75	1.00	0.86	**1.00**	1.00	1.00	1.00	**0.80**	0.75	1.00	0.86
Support Vector Classifier	**0.80**	0.75	1.00	0.86	0.60	0.67	0.67	0.67	0.60	0.67	0.67	0.67	0.60	0.67	0.67	0.67	**0.80**	0.75	1.00	0.86
Naive Bayes	**0.80**	0.75	1.00	0.86	0.60	0.60	1.00	0.75	**0.80**	0.75	1.00	0.86								
AdaBoost	**0.80**	0.75	1.00	0.86	0.57	0.50	1.00	0.67	0.60	0.67	0.67	0.67	**1.00**	1.00	1.00	1.00	**0.80**	0.75	1.00	0.86
Gradient Boost	**0.80**	1.00	0.67	0.80	0.80	0.75	1.00	0.86	**0.80**	0.75	1.00	0.86	**1.00**	1.00	1.00	1.00	**0.80**	0.75	1.00	0.86
Linear SVC	**0.80**	0.75	1.00	0.86	0.80	1.00	0.67	0.80	**0.80**	1.00	0.67	0.80	0.80	1.00	0.67	0.80	**0.80**	1.00	0.67	0.80

[1] The highest values across all feature sets are given in bold, A: Accuracy, P: Precision, R: Recall, F: F1 score

Table A6. Performance metrics for Video 6 per feature set (R – Raw, E – Engineered, FR() - feature reduction)

Algorithms	R				E				R+E				FR(R+E)				R+FR(E)			
	A	P	R	F	A	P	R	F	A	P	R	F	A	P	R	F	A	P	R	F
Logistic Regression	0.40	0.25	1.00	0.40	0.60	0.33	1.00	0.50	**0.60**	0.33	1.00	0.50	0.60	0.33	1.00	0.50	0.60	0.33	1.00	0.50
KNN classifier	0.20	0.20	1.00	0.33	0.60	0.33	1.00	0.50	**0.60**	0.33	1.00	0.50	0.60	0.33	1.00	0.50	0.60	0.33	1.00	0.50
Random Forest	**0.80**[1]	0.50	1.00	0.67	0.60	0.33	1.00	0.50	**0.60**	0.33	1.00	0.50	0.60	0.33	1.00	0.50	**0.80**	0.50	1.00	0.67
Support Vector Classifier	0.20	0.20	1.00	0.33	0.60	0.33	1.00	0.50	0.20	0.20	1.00	0.33	0.60	0.33	1.00	0.50	0.40	0.25	1.00	0.40
Naive Bayes	0.40	0.25	1.00	0.40	0.20	0.20	1.00	0.33	0.40	0.25	1.00	0.40								
AdaBoost	0.60	0.00	0.00	0.00	0.43	0.20	1.00	0.33	**0.60**	0.00	0.00	0.00	0.60	0.33	1.00	0.50	0.60	0.00	0.00	0.00
Gradient Boost	0.40	0.00	0.00	0.00	0.60	0.33	1.00	0.50	**0.60**	0.33	1.00	0.50	0.40	0.25	1.00	0.40	**0.80**	0.50	1.00	0.67
Linear SVC	0.40	0.25	1.00	0.40	**0.80**	0.50	1.00	0.67	**0.60**	0.33	1.00	0.50	**0.80**	0.50	1.00	0.67	0.60	0.33	1.00	0.50

[1] The highest values across all feature sets are given in bold, A: Accuracy, P: Precision, R: Recall, F: F1 score

To cite this article: Karasavvidis, I., Papadimas, C., & Ragazou, V. (2022). Student-generated texts as features for predicting learning from video lectures: An initial evaluation. *Themes in eLearning*, 15, 21-45.

URL: http://earthlab.uoi.gr/tel