Classification of pedagogical content using conventional machine and deep learning model

Vedat Apuk, Krenare Pireva Nuci
Department of Computer Science and Engineering
University for Business and Technology
10000 Prishine, Kosovo
Email: krenare.pireva@ubt-uni.net

Abstract—**The advent of the Internet and a large number of digital technologies has brought with it many different challenges. A large amount of data is found on the web, which in most cases is unstructured and unorganized, and this contributes to the fact that the use and manipulation of this data is quite a difficult process. Due to this fact, the usage of different machine and deep learning techniques for Text Classification has gained its importance, which improved this discipline and made it more interesting for scientists and researchers for further study. This paper aims to classify the pedagogically content using two different models, the K-Nearest Neighbor (KNN) from the conventional models and the Long short-term memory (LSTM) recurrent neural network from the deep learning models. The result indicates that the accuracy of classifying the pedagogical content reaches 92.52 % using KNN model and 87.71 % using LSTM model.**

Keywords: Document Classification, KNN, LSTM, coursera dataset, education, text classification, deep learning models, machine learning models

I. Introduction

Billions of users create a large amount of data every day which in a sense comes from various types of sources. This data is in most cases unorganized and unclassified and is presented in various formats such as text, video, audio, or images.

Processing and analyzing this data is a major challenge that we face every day. The problem of unstructured and unorganized text dates back to ancient times, but Text Classification as a discipline first appeared in the early 60s, where 30 years later the interest in various spheres for it increased [1], and began to be applied in various types of domains and applications such as for movie review [2], document classification [3], ecommerce [4], social media [5], online courses [6, 7].

As interest has grown more in the upcoming years, the uses start solving the problems with higher accurate results in more flexible ways. Knowledge Engineering (KE) was one of the applications of text classification in the late 80s, where the process took place by manually defining rules based on expert knowledge in terms of categorization of the document for a particular category [1]. After this time, there was a great wave of use of various modern and advanced methods for text classification, which all improved this discipline and made it more interesting for scientists and researchers, more specifically the use of machine learning techniques. These techniques bring a lot of advantages, as they are now in very large numbers, where they provide solutions to almost every problem we may encounter.

The need for education and learning dates back to ancient times, where people are constantly improving and trying to gain as much knowledge as possible. There are various sources of learning available today including various MOOC platforms such as Coursera, Khan Academy, Udemy, Udacity, edX, to name a few, and as technology has evolved it has contributed to better methods of acquiring knowledge that will facilitate this process. The data coming from these sources are in most cases in digital form, more specifically in the form of video and text lessons. The platforms that contain these lessons are called Massive Open Online Courses (MOOCs), where in addition to the video lesson, it also contains its textual representation called a transcript. Considering that the duration of a video lesson depends on several parameters, such as the category of video material, the platform on which the lesson is provided, the complexity of the topic, the number of instructors, and the group of lesson attendants. The duration of the lessons indirectly dictates how long the transcript will be, in other words how many words it can contain. The category shows the nature of the video and the topics that will be presented in it. As it is already known, that each video lesson belongs to a certain category, or in a group of categories, so does the transcript as well. Given this advantage, we can conclude the fact that text classification is becoming quite extensive as a discipline, where also its use can solve many challenging problems in every domain, and specifically in education domain.

The aim of this paper is to investigate two classification techniques that are used to classify the pedagogical content, and the focus tends to compare conventional machine learning models with deep learning models, by selecting KNN algorithm for the first approach and LSTM architecture for the latter one.

To better indicate the idea we want to present, the paper will be divided into several sections, as follows: as part of literature review the main processes of classifying documents are explained, continuing with related work conducted so far in this area. In the experimental section, the design of the conventional machine learning models and deep learning models will be elaborated and the results for the each of the architectures will be presented using a number of evaluation techniques (recall, precision, F-Score, accuracy). The paper will be concluded with conclusion and future work.

II. RELATED WORK

Text mining or text analytics is one of the artificial intelligence techniques that uses Natural Language Processing (NLP) to transform unorganized and unstructured text into an appropriately structured format that will make it easier to process and analyze data. For businesses and other corporations, generating large amounts of data has become a daily routine. Analysis of this data help companies gain smarter and more creative insights regarding their services or products collected from a variety of sources in automated manner. But this analysis step requires processing a huge amount of data where the data needs to be prepared, and this is in most cases the cause of various problems.

NLP consists of five steps or phases (see Figure 1), such as: Lexical Analysis, Syntax Analysis, Semantic Analysis, Pragmatics, and Discourse [8].

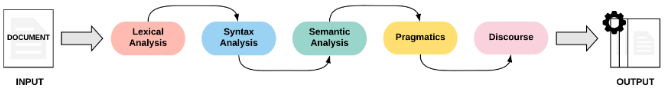

Figure 1: Natural Language Processing steps.

So, the goal of text classification or text analysis is to structure and classify data to facilitate the analysis process. Today, as shown in Figure 2, in order to perform text classification in the existing data, we follow the four phases [9]:

a) *Feature Extraction,*
b) *Dimension Reductions,*
c) *Classifier Selection,*
d) *Evaluation.*

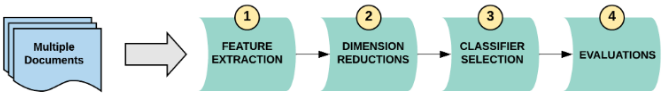

Figure 2: Four-phase model of a text classification system.

As shown in Figure 2, with feature extraction as an initial phase one piece of text or document is converted into a so-called structured feature space, which will be useful to us when using a classifier. But prior to this, needs to perform data cleaning, taking care of missing data, removal of unnecessary characters or letters, in order to bring the data in an appropriate shape for extracting the features, otherwise omitting the data cleaning can directly affect negatively the performance and the accuracy of the final results.

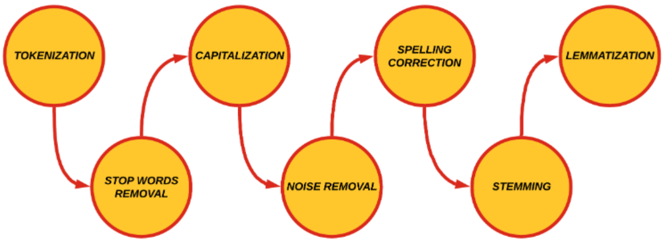

Figure 3: Techniques of data preprocessing phase.

Emphasizing the importance of pre-processing data, in Figure 3, are depicted a number of processes that are followed to clear the data and prepare it for further processing [9]. Such processes as:

- Tokenization - is the process of separating a piece of text into smaller units called tokens. The way the token isformed is based on a delimiter, which in most cases is space. Also, tokens can be words or sub-words, but alsoat a lower level, based on characters.
- Stop Words - are words that are commonly used in one language, that are unnecessary in the data processingpart, and in most cases are ignored because they take up more space in the database, and affect longerprocessing times. In English stop words are words like: "a", "the", "an", "it", "in", "because", "what", to namea few.
- Capitalization - is the part where it is necessary to identify the correct capitalization of the word, where thefirst word in the sentence will be automatically capitalized first.
- Noise Removal - is the process of removing characters, numbers, and parts of text that affect your analysis.These characters can be some special characters, punctuation, source code removal, HTML code removal,unique characters that represent a particular word, numbers, and many other identifiers.
- Spelling Correction - is a problem where the meaning of a particular word can be mispronounced, where theword loses its meaning. This problem can be solved in two ways: with edit distance and another with overlapusing k-gram.
- Stemming - is a process where more morphological variants are produced than the base word or the so-calledroot word. For example different morphological variants of root words "like" such as "likes", "liked", "liking"and "likely".
- Lemmatization - in this technique words are replaced with root words or words that have a similar meaning,and such words are called lemmas.•Syntactic Word Representation (such as N-Gram) - is a contiguous sequence of n items from one part of thetext.• Syntactic N-Gram - are n-grams that are constructed using paths in syntactic trees.–Weighted Words (such as TF and TF*IDF)–Word Embedding (such as Word2Vec, GloVe, FastText)

After finalizing with data pre-processing step, we continue with Dimension Reductions. With dimension reductions we transform the data from a high-dimensional space to a low-dimensional space. The reason for this is that we strive to improve performance, speedup time, and reduce memory complexity. As shown in Figure 4, there are a number of algorithms or techniques in this step that could be implemented, such as: (i) Principal Component Analysis (PCA), (ii) Non-negative Matrix Factorization (NMF), (iii) Linear Discriminant Analysis (LDA)and (iv) Kernel PCA.

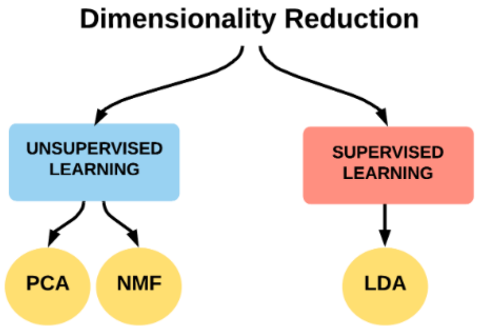

Figure 4: Categorization of dimension reductions algorithms

As part of four-phase model of a text classification system depicted in Figure 2, in the pre-final phase we deal with classifier selection. One of the main concerns is to choose the right classifier model that will be able to perform with a certain set of data to achieve the desired results. Choosing the right classifier model is not an easy task, and is a challenge that is also referred to in the literature as the Algorithm Selection Problem (ASP). Every day we come across applications that use classification algorithms in some hands. The results of the task depend on choosing the right algorithm that will complete a particular job while showing very good performance and problem optimization. In general, there is no single algorithm that can work for every type of problem, and that can learn all the tasks while still being efficient, and this phenomenon is also known as performance complementary [10]. Many factors affect the performance of a particular algorithm, some of which is the amount of data assigned to it for testing and training, the operating system to be executed, the specifications of the machine on which the algorithm will be performed, and many other factors that directly or indirectly affect the selection of the algorithm. Some of the algorithms used for text classification are: Logistic Regression, Naive Bayes, K-Nearest Neighbor (KNN), Support Vector Machines (SVM), Decision Trees, Random Forests, Neural Network algorithms (such as DNN, CNN, RNN) and Combination Techniques. In our experiment we have used K-Nearest Neighbor (KNN) from the conventional models whereas LSTM recurrent neural network from the deep learning models. To conclude, the evaluation phase is encountered as the final step when creating a model for text classification is the evaluation phase. In this phase, algorithms are analyzed or scored to assess how efficiently they performed. Today, the various technologies available today have drastically improved the way people try to gain new knowledge. Technology has greatly influenced the improvement of this process, and at the same time contributed to the development of systems that enable a more efficient and easier learning process. With this fact the use of various Massive Open Online Courses (MOOCs) begins to increase, which bring with them various opportunities, but also challenges. Attempts to identify and analyze the opportunities and challenges of MOOCs both from pedagogical and business standpoint have led to understand how some of the very well known and successful platforms like Coursera, edX and Udacity have contributed to the improvement of their business model through various aspects, using the models for: certification model, freemium model, advertising model, job-matching model, and subcontractor model [8]. During the analysis of these platforms, the authors in [9] concluded that quite a low number of students actually take assessment exams at the end of a MOOC which makes it difficult to assess whether students joining a MOOC are actually learning the content, and hence whether the MOOC is achieving its goal. One of the main components of these platforms is Learning Objects (LOs). Various techniques regarding Learning Objects (LOs) representation are presented, in which it contains pedagogical values [10].

Using the representation features of Learning Objects will provide possibilities to personalize and customizable contents when presenting to learners along with the ability to choose an individual learning path that best suits them, aiming to maximize the learning outcome as claimed in [10]. There are plenty of examples where K-Means, Decision Trees, Deep Neural Network (DNN) and other machine learning techniques have been used for classification purposes [11]. As eLearning platforms are becoming more accessible, where their main goal is to provide a smarter way of learning. The new paradigm of e-Learning also known as Cloud eLearning aiming to offer personalised learning using Cloud resources, where the main challenge is the process of content classification and matching it with learners preferences. As part of this work, the author [12] integrated as middle layer the recommendation systems using hierarchical clustering technique to recommend learners courses or materials that are similar to their needs before proposing a learning path using artificial intelligent automated planner. Also, paper [13] contributes to the classification systems in pedagogical content, with the main focus on the content classification of video lectures. The authors recommended model for the visual content classification system (VCCS) for multimedia lecture videos is to classify the content displayed on the blackboard. Through this recommended model, the authors showed over several stages how lecture videos are processed and then with a combination of support vector machines (SVM) and optical character recognition (OCR) classifies visual content, text and equations [13]. Furthermore in [14], researchers presented the classification and organization of pedagogical documents using domain ontology.

In one of the previous studies [15], the authors of this paper presented a technique for automatic classification of MOOC videos, where the first step is to extract transcripts from video and then convert them into image representation using a statistical co-occurrence transform. After that, a CNN model with a dataset was implemented which was collected from Khan Academy with a total of 2545 videos, in order to evaluate the technique presented in the paper. Based on label accuracy, the best results were achieved with the CNN model, with the value of 97.87%. Also, similar work has been carried out in [16] where they have proposed a video classification framework, consisting of three main modules: pre-processing, transcript representation, and classifier. In this paper, it was concluded that much better classification results were achieved with general-level than with specific-level, argued with the fact of class overlap that the specific-level category contains.

This paper aims to classify the pedagogical content using two different algorithms, K-Nearest Neighbour as a conventional machine learning model and Long short-term memory (LSTM) as an artificial recurrent neural network architecture used in deep learning.

III. METHODOLOGY

In this section is given the methodology used during the research and the experimental part. Initially a brief introduction regarding the dataset is given, and continuing with explanation of the architectures that are modelled to classify pedagogical content.

Python technology is used for the whole experiment, and specifically to implement the KNN model is used the built-in functions and modules of scikit-learn library, whereas for the implementation of the RNN model is used Keras library, that runs on top of Tensorflow. In the following subsections, the used dataset as part of this experiment is described in detail, following with both models, the KNN and LSTM.

A. *Dataset*

The process of collecting and reviewing data is not an easy task, and in most cases requires a lot of research and finding relevant data that are used to achieve the desired results. The dataset [17] used in this paper for the experimental purposes is used in [16] and it is modelled from scratch. This dataset consists of a total of 12,032 videos collected from the Coursera platform from more than 200 different courses. Coursera categorizes courses into a 2-level hierarchical structure from general level to fine-grained level. The general level consists of 8 categories, the specific level of 40 categories, and the course level of a total of 200 categories. In addition to these three levels that made up the course, a video lesson transcript was also included.

Figure 5 presents the top five most frequent categories, while Figure 6 presents the top five least frequent categories by the number of transcripts that these categories contained. In order for the data to be in the correct format for further analysis and modeling process, the data needs to go under pre-process phase, by preparing, cleaning, and transformed in a desired shape.

The data preparation and preprocessing part depends on the given dataset, and in our case the first step after the review is to remove the noisy data (such as '[MUSIC]' which are recorded very frequently in all transcript records). Following the steps depicted in Figure 3, the entire textual content of the transcript is converted into lowercase, and removed the nonletters characters. Further, the stopwords are removed from the transcript where it helped us reduce the derived words to their particular word stem or root word.

The dataset is transformed finally into the desired shape after finishing the lemmatization process, and it is ready to be used for both architectures that we have modelled, KNN and LSTM described further in the following subsections.

B. *K-Nearest Neighbour model*

K-Nearest Neighbors (KNN) is one of the techniques that is used in both classification and regression. It is known that KNN has no model other than collecting the entire dataset, and there is no need for learning. The predictions made with the KNN for the new data point are by searching the entire dataset for the K most similar instance (so-called neighbors) in relation to the output variant of the K instance [18].

There are a number of steps that the KNN algorithm goes through, such as:

1) *Modify K with the number of specific neighbors.*
2) *Calculate the distance between the available raw data examples.*
3) Sort the calculated distances.
4) Get the labels of top K entries.
5) Generated prediction results for the test case.

In this experiment, while implementing the KNN model, immediately after the process of cleaning and preparing data, is built a dictionary of features, which transforms documents to feature vectors and convert the transcripts of documents to a matrix of token counts using CountVectorizer method.

Then, the count matrix is transformed to a normalized tf-idf representation using Tfidf transformer method. After this is identified the exact number of neighbors which in our case resulted in 7 neighbors. To train the classifier, the dataset is divided into two subsets: 80% for training and 20% for testing. Where the latter subset is used to predict the category for each input text record.

C. Long short-term memory model

Recurrent Neural Networks (RNN) are types of artificial neural networks that allow previous outputs to be used as inputs while having hidden states [19]. These algorithms are mostly used in fields such as: Natural Language Processing (NLP), Speech Recognition, Robot Control, Machine Translation, Music Composition, Grammar Learning, and many others. Typically, a feedforward network maps one input to one output. But as such, the inputs and outputs of neural networks can vary in the length and type of networks used for different examples and applications [20].

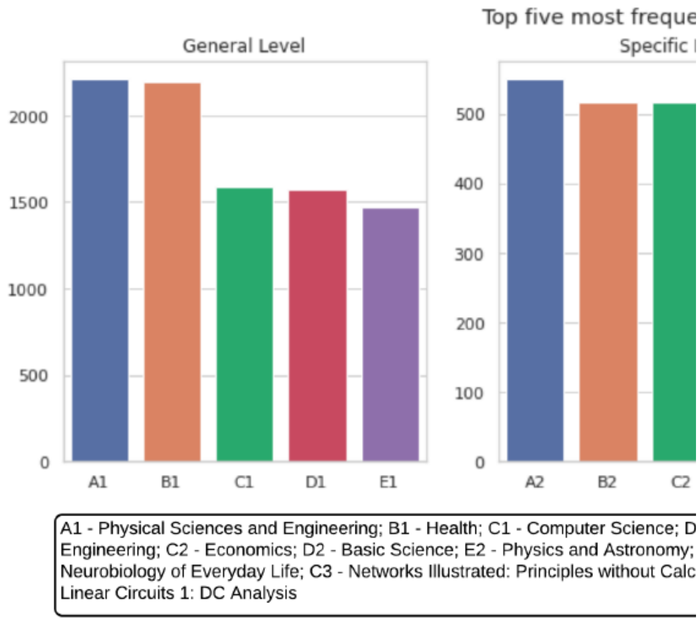

Figure 5: Top five most frequent categories for all three levels.

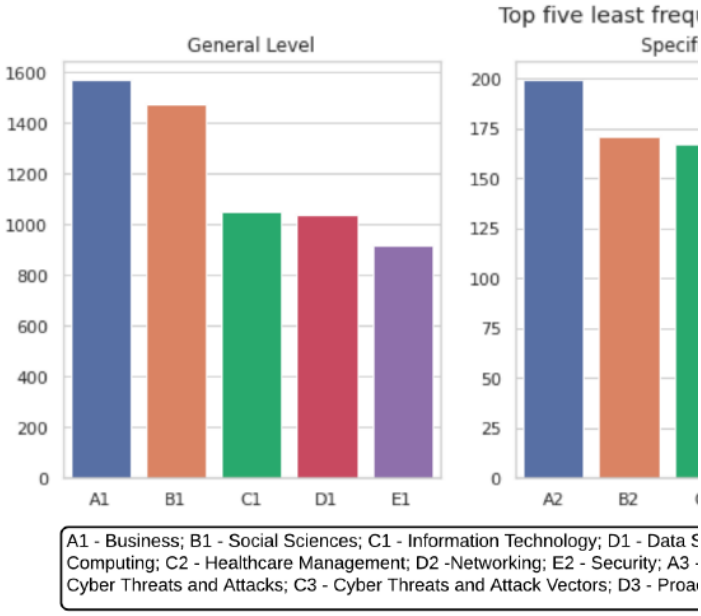

Figure 6: Top five least frequent categories for all three levels

In the implementation of our LSTM model, in order to implement the RNN model, we used the LSTM architecture that remembers values over arbitrary intervals. As part of this architecture firstly are created sequence models as the input layer to our network, then adding the Embedding layer which encodes to integer values the textual data entered as input, and as a result of this layer each word is then represented by a unique integer. For this layer, we have specified three required parameters with their respected values:

- Maximum number of words - which in our case is 50000.
- Embedding Dim - 100.
- Input length - shape of X value which for us is 3002.

Further are dropped out hidden and visible units between the layers in the network, with a dropout rate of 0.2, the same value is for recurrent dropout as well. This is followed by the implementation of LSTM layer, and Dense layer to which we passed as the first parameter the number of units denoting the dimensionality of the output space, which in our case depends on the number of categories that are selected to classify, and as the second parameter the activation function, in this case is chosen the softmax function. And as a final step, is used categorical crossentropy as a loss function, and Adam as an optimizer of the network. To prevent underfitting or overfitting of the network, and to select the appropriate number of training epochs is used EarlyStopping with 'val loss' as a monitoring metric with patience of 3 epochs.

IV. RESULTS

Table I shows the classification results with the conventional model using K-Nearest Neighbours algorithm. As shown in Table I, the general level based on the precision metric has shown a very good result, 92.63% of accuracy whereas 87.89% accuracy is estimated by precision metrics specific level. And at the course level, also based on the precision metric reaches 78.59%. Analyzing the results for all three levels, we notice that the percentage of accuracy decreases from the upper level (general level) up to the lower level (course level). In our case, the general level consists of 8 sub-categories, the specific level of 40 sub-categories, and the course level consists of 200 subcategories.

From this we can infer that that the number of subcategories for a single level by which the video is classified on the Coursera platform differs in each level.

TABLE I. CLASSIFICATION RESULTS WITH K-NEAREST NEIGHBOURS

Category	***Precision (%)***	***Recall (%)***	***F1 Score (%)***	***Accuracy (%)***
General Level	92.63	92.52	92.53	92.52
Specific Level	87.89	87.58	87.49	87.58
Course Level	78.59	76.73	76.11	76.73

Table II shows the classification results with the Recurrent Neural Networks, more specifically with an Long Short-Term Memory (LSTM) architecture. Using LSTM classifier, the general level based on the precision metric reaches 88.22% of accuracy whereas in the specific level, 72.31%. Finally, at the course level, the results shows 59.49% of accuracy. Analyzing the results using LSTM architecture the highest accuracy is achieved at the general level, followed by a specific level, while the lowest accuracy is achieved at the course level.

TABLE II. CLASSIFICATION RESULTS WITH RECURRENT NEURAL NETWORKS

Category	***Precision (%)***	***Recall (%)***	***F1 Score (%)***	***Accuracy (%)***
General Level	88.22	87.71	87.68	87.71
Specific Level	72.31	69.93	70.13	69.93
Course Level	59.49	52.91	53.99	52.91

V. CONCLUSION AND FUTURE WORK

In this paper are presented and discussed the classification results of the conducted experiment for all three category levels (General, Specific and Course level) using both architectures, KNN and LSTM. We can conclude that better results are achieved for levels with a smaller number of categories than for levels with a larger number of categories. In our case, as the category number increased in classes the results decreased. With this, we claim that the classification results are directly affected by the number of categories that each level contains. From results shown in Table I and Table II KNN reached 92.52% of accuracy compared to LSTM with 87.71% at general level, 87.58% compared to 69.93% at specific level and finally 76.73% compared to 52.91% at course level. The conducted results could be affected from several factors. First, the quantity of data required for LSTM, since a large number of categories increases the complexity of the problem, and thus requires more data to train the model. The result could have been affected due to the high similarity of different transcripts. Many of the transcripts belonged to different classes at the course level, and they had many similarities in the context of the sentences and keywords, so the model could not properly distinguish in which class the transcripts belonged. However, the final results gives us a spark for future work to investigate more on recurrent neural networks like, applying hyperparameters tuning, or even expand the number of architectures to further investigate the pedagogical content classification.

REFERENCES

[1] Fabrizio Sebastiani. Machine learning in automatedtext categorization.ACM computing surveys(CSUR), 34(1):1–47, 2002.

[2] Cicero Dos Santos and Maira Gatti. Deep convo-lutional neural networks for sentiment analysis ofshort texts. InProceedings of COLING 2014,

the25th International Conference on ComputationalLinguistics: Technical Papers, pages 69–78, 2014.

[3] Zenun Kastrati, Ali Shariq Imran,and Sule Yildirim Yayilgan. A general frameworkfor text document classification using semconand acvsr. InInternational Conference on Human Interface and the Management of Information,pages 310–319. Springer, 2015.

[4] Arfinda Ilmania, Samuel Cahyawijaya, Ayu Pur-warianti, et al. Aspect detection and sentimentclassification using deep neural network for in-donesian aspect-based sentiment analysis.In2018 International Conference on Asian LanguageProcessing (IALP), pages 62–67. IEEE, 2018.

[5] Ali Shariq Imran, Sher Muhammad Daudpota,Zenun Kastrati, and Rakhi Batra. Cross-culturalpolarity and emotion detection using sentimentanalysis and deep learning on COVID-19 relatedtweets. IEEE Access, 8:181074–181090, 2020.

[6] Zenun Kastrati, Ali Shariq Imran, and Arianit Kurti.Weakly supervised framework for aspect-basedsentiment analysis on students' reviews of MOOCs.IEEE Access, 8:106799–106810, 2020.

[7] Alya Itani, Laurent Brisson, and Serge Garlatti.Understanding learner's drop-out in MOOCs. Ininternational conference on intelligent data engi-neering and automated learning, pages 233–244.Springer, 2018.

[8] Hannes Max Hapke, Hobson Lane, and ColeHoward. Natural language processing in action,2019.

[9] Kamran Kowsari, Kiana Jafari Meimandi, MojtabaHeidarysafa, Sanjana Mendu, Laura Barnes, andDonald Brown. Text classification algorithms: Asurvey. Information, 10(4):150, 2019. [10] Irfan Khan, Xianchao Zhang, Mobashar Rehman,and Rahman Ali. A literature survey and empiricalstudy of meta-learning for classifier selection.IEEEAccess, 8:10262–10281, 2020.

[10] Jake Lever, Martin Krzywinski, and Naomi Altman.Erratum: Corrigendum: Classification evaluation.Nature Methods, 13(10):890–890, 2016.

[11] Fisnik Dalipi, Sule Yayilgan, Ali Shariq Imran, andZenun Kastrati. Towards understanding the MOOCtrend: pedagogical challenges and business oppor-tunities. InInternational Conference on Learning and d Collaboration Technologies, pages 281–291.Springer, 2016.

[12] Krenare Pireva, Ali Shariq Imran, and FisnikDalipi.User behaviour analysis on LMS andMOOC. In2015 IEEE Conference on e-Learning,e-Management and e-Services (IC3e), pages 21–26.IEEE, 2015.

[13] Fisnik Dalipi, Ali Shariq Imran, and Zenun Kastrati.MOOC dropout prediction using machine learningtechniques: Review and research challenges. In2018 IEEE Global Engineering Education Confer-ence (EDUCON), pages 1007–1014. IEEE, 2018.

[14] Krenare Pireva and Petros Kefalas. A recommendersystem based on hierarchical clustering for cloude-learning. InInternational Symposium on Intel-ligent and Distributed Computing, pages 235–245.Springer, 2017.

[15] Ali Shariq Imran and Faouzi Alaya Cheikh. Black-board content classification for lecture videos. In2011 18th IEEE International Conference on ImageProcessing, pages 2989–2992. IEEE, 2011.

[16] Ali Shariq Imran and Zenun Kastrati. Pedagogicaldocument classification and organization usingdomain ontology. InInternational Conference onLearning and Collaboration Technologies, pages499–509. Springer, 2016.

[17] Houssem Chatbri, Kevin McGuinness, SuzanneLittle, Jiang Zhou, Keisuke Kameyama, Paul Kwan,and Noel E O'Connor. Automatic mooc videoclassification using transcript features and convolu-tional neural networks. InProceedings of the 2017ACM Workshop on Multimedia-based Educationaland Knowledge Technologies for Personalized andSocial Online Training, pages 21–26, 2017.

[18] Zenun Kastrati, Ali Shariq Imran, and ArianitKurti. Integrating word embeddings and documenttopics with deep learning in a video classificationframework.Pattern Recognition Letters, 128:85–92, 2019.[

[19] Zenun Kastrati, Arianit Kurti, and Ali Shariq Imran.Wet: Word embedding-topic distribution vectorsfor mooc video lectures dataset.Data in brief,28:105090, 2020.

[20] Jason Brownlee.Master Machine Learning Al-gorithms: discover how they work and implementthem from scratch. Machine Learning Mastery,2016.

[21] Afshine Amidi and Shervine Amidi. Vip cheatsheet:Recurrent neural networks, 2018.

[22] Larry Medsker and Lakhmi C Jain.Recurrentneural networks: design and applications. CRCpress, 1999