Winter 1-2021

CLASSIFICATION OF PEDAGOGICAL CONTENT: REVIEW AND RESEARCH CHALLENGES

Vedat Apuk

Programi për Shkenca Kompjuterike dhe Inxhinierise

CLASSIFICATION OF PEDAGOGICAL CONTENT: REVIEW AND RESEARCH CHALLENGES

Shkalla Bachelor

Vedat Apuk

Janar / 2021
Prishtinë

Programi për Shkenca Kompjuterike dhe Inxhinierise

Punim Diplome
Viti akademik 2017– 2018

Vedat Apuk

CLASSIFICATION OF PEDAGOGICAL CONTENT: REVIEW AND RESEARCH CHALLENGES

Mentori: PhD Krenare Pireva Nuçi

Janar / 2021

Ky punim është përpiluar dhe dorëzuar në përmbushjen e kërkesave të pjesshme për Shkallën Bachelor

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 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. 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. With this, we can notice that text classification is quite extensive as a discipline. The objective of this paper is to indicate several different classification techniques that will classify a transcript from a video lesson into the specific category to which it belongs.

ACKNOWLEDGEMENTS

Writing this thesis paper was very challenging but at the same time very inspiring and proud. The overall road was not easy and this thesis is the result of hard work, knowledge, sacrifice, and positivity. However, the journey and the results would not be possible without unconditional support, love, and trust from several people.

Firstly, I want to thank God, for providing me His countless blessings throughout this overall research journey, in order to complete it successfully.

Secondly, I would like to express my sincere and deep gratitude and appreciation to my thesis supervisors, Krenare Pireva Nuçi and Ali Shariq Imran, for providing me with invaluable guidance and support throughout the research process. Their unconditional support, dynamism, sincerity and clear vision, motivation, and sincerity have deeply and constantly inspired me. They showed me the methodology and the process to carry out this type of research and to present its results as clearly as possible. It was my great honor and privilege to learn and work under their supervision and guidance. I am extremely grateful for this opportunity.

Furthermore, I am extremely thankful and grateful to my parents, for their support, prayers, caring, love, and sacrifices for preparing and educating me for my better future. Also, I am very thankful to my brothers for their tremendous support, understanding, valuable prayers, and love. I would like to express my appreciation to my close friends for their genuine support.

I am extending my thanks to UBT who provided me an opportunity to complete my final semester in Sweden at Linnaeus University, to further extend and advance my knowledge, skills, and professional experience abroad.

Finally, my thanks and huge gratitude goes to all the people who have continuously and undoubtedly supported and believed in me to complete the research paper indirectly and directly. Without you, nothing would be possible. This one's for you.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY

NLP - Natural Language Processing

KE - Knowledge Engineering

TF - Term Frequency

TF*IDF – Term Frequency-Inverse Document Frequency

PCA – Principal Component Analysis

NMF – Non-negative Matrix Factorization

LDA – Linear Discriminant Analysis

ASP – Algorithm Selection Problem

KNN – K-Nearest Neighbor

SVM – Support Vector Machines

DNN – Deep Neural Network

CNN – Convolutional Neural Network

RNN – Recurrent Neural Network

LSTM - Long Short-Term Memory

1. 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 in the early ‘90s interest in various spheres for it increased [1], and began to be applied in various types of domains and applications. As interest has grown more and the years, where the uses of these applications have begun to solve problems that allow for easier and more flexible ways to arrive at more accurate results. 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, 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 lessons. The platforms that contain these video 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. From this advantage, we can conclude the fact that text classification is quite extensive as a discipline, where also its use can solve many challenging problems. To better indicate the idea we want to present, the paper will be divided into several sections, as follows: Chapter 2 presents a declaration of the problems and challenges of document classification, as well as the objectives of the paper. Chapter 3 will explain the process of classifying documents, classification techniques, applications and domains of its use, as well as previous works and research that have been done in this area. The following is Chapter 4, which will present the methodology of work and the objectives of studying this paper. Chapter 5 focuses on the design and implementation of the experiment, followed by Chapter 6 where the results of the experiment will be presented and discussed. And the last Chapter 7 shows the conclusions of the results of the experiments, and this paper in general.

2. PROBLEM DECLARATION AND OBJECTIVES

As we stated in the previous chapter, where the need for education is growing, the development of technology in the provision of better services has had a significant impact. One of the methods that has helped a lot to make it easier and more efficient for people to access educational resources is the MOOCs platform. As the name suggests it is that these platforms are designed to support a large number of participants. Given this fact, the number of courses and materials on these platforms has increased dramatically in recent years. As already mentioned, the video lessons on these platforms also contain a transcript. And as such, this fact shows that a large number of video lessons already exist on these platforms, and their classification by certain techniques is not an easy task. The aim of this paper is to indicate several different classification techniques that will classify a transcript from a video lesson into the specific category to which it belongs. One of the challenges in selecting classifiers is to adapt to the number of classes that may be available. In most cases, there is more than one category, where the use of single-class or so called one-class classifiers is not a good idea, so the use of multi-class classifiers is needed, but it is still a challenge in predicting the accuracy of the results. In this paper, we will select techniques and create models for multi-class classification. And after that, we will compare the results of both techniques, and come to the conclusion which of them better classifies in the case when a large number of classes are involved. For the automatic classification of the documents in this paper we will use supervised algorithms.

There are a total of six main objectives that are part of this paper, and they are as follows:

> ***Objective 1:*** Explore several techniques for both single-class and multi-class classification.
>
> ***Objective 2:*** Review different datasets that best fit the case for classification and are from the category of pedagogical domain. These datasets can be provided by various MOOCs platforms.

Objective 3: Select two appropriate multi-class classification techniques.

Objective 4: Create a model using both classification techniques.

Objective 5: Compare the results of these previously created models.

Objective 6: Design scenarios to test the models.

3. LITERATURE REVIEW

This chapter will try to present previous work on the topic of classification of documents for pedagogical content, and the challenges that come with it. Also, the concepts of document classification, classification techniques and applications and domains in which it is used will be pointed out. Each of the above topics will be explained in the remainder of the chapter.

3.1 Background

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 gains smarter and more creative insights regarding their services or products collected from a variety of sources. 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. We can point out that NLP is one of the analysis methodologies used in text mining, and also depending on what kind of approach is used by this methodology.

Text mining and NLP are closely related to each other, where NLP can help machines to understand natural languages spoken by human beings, like English or any other language. In some other words, it can be described as a concept of creating and understanding expressions in human language, or so-called natural language. And as such NLP is made up of five steps or phases, and they are *Lexical Analysis*, *Syntax Analysis*, *Semantic Analysis*, *Pragmatics,* and *Discourse* [2].

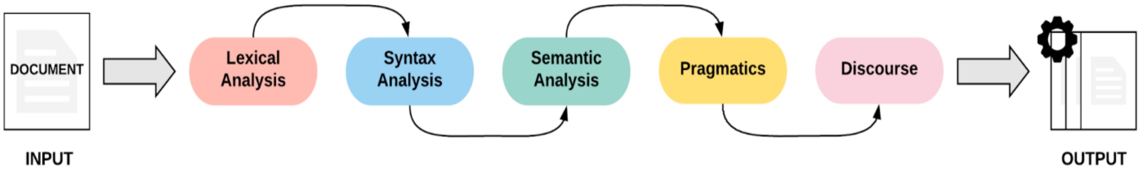


Figure 1. Natural Language Processing steps.

Figure 1 shows the steps of NLP, where each of these steps will be briefly described, with the idea of a better understanding of how this methodology works. The following is a description of these steps:

1. ***Lexical Analysis*** - involves identifying the structure of a sentence, to separate words from the text, and create individual words, sentences, or paragraphs, which also includes separating punctuation from words
2. ***Syntax Analysis*** - involves parsing words and arranging words in a sentence to have a certain meaning and relationship between them, where it is based exclusively on grammar.
3. ***Semantic Analysis*** - implicates to analyze the grammatical structure of a word and seeks for a specific meaning in that word. The semantic analysis makes it possible to understand the relationship between lexical items.
4. ***Pragmatics*** - means how the interpretation of a sentence is affected in its use in different situations to understand what it means and encompasses.
5. ***Discourse*** - points out that the current sentence may depend on the previous sentence, where it can also affect the meaning of the sentence that comes after it.

Visualization and description of the steps of NLP are aimed at giving the reader more knowledge about this methodology or even its refreshment which will be helpful in the following sections of the paper.

As already stated above, the goal of text classification or text analysis is to structure and classify data to facilitate the analysis process. And just like many other smaller tasks or sub-processes that make up this overall flow of steps or so-called text classification pipeline. It can be observed that text classification systems can be found presented in various scientific papers, where researchers have contributed to several types of division and the number of steps that make up text classification systems. These divisions imply how one process will be presented, and whether it needs to be divided into one bigger step or a few smaller steps, but the overall meaning of the separation is the same in almost every paper.

In this paper, we will present the four phases of which most text classification systems consist [3], and they are:

I. Feature Extraction
II. Dimension Reductions
III. Classifier Selection
IV. Evaluation

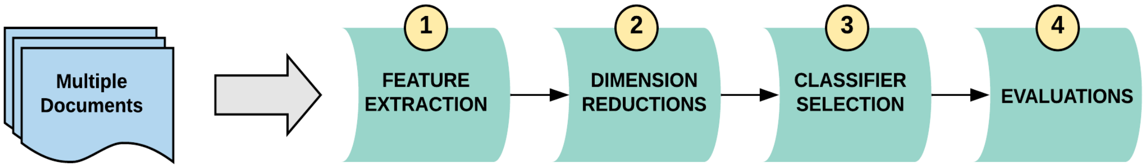


Figure 2.Four-phase model of a text classification system.

Figure 2 shows a four-phase model of a text classification system, where we tried to explain it as clearly as possible every detail that is important in the text classification. The explanation of each phase is shown in the next section.

3.2 Related Work

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. Through SWOT analysis have been presented strengths, weaknesses, opportunities and threats related to MOOCs. 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 [4]. Each of these platforms strives to improve the business model and strategy, in order to contribute to better results, and most importantly increase user interest. During the analysis of these platforms it was 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. We can observe five common business models of these platforms, which are: certification model, freemium model, advertising model, job-matching model, and subcontractor model. The main goal of these models and their provision is charging for certificates, linking students with potential employers, and charging for additional services [4].The Learning Management System (LMS) is a platform that supports and hosts MOOCs. In addition to LMS, one of the components of the e-learning system architecture is Learning Objects (LOs). Various techniques regarding

Learning Objects (LOs) are presented, in a context in which it contains pedagogical values. These LOs may contain data that is in digital and non-digital form. Referring to paper [5], the authors have identified several very important features that are not present in existing LMSs, and believe that they should be integrated into these LMSs. This will increase collaboration and interaction between users and course content. The author's recommendations are: personalization of learning path, customizable video learning objects (VLO), ontology adaptation, multi-agent systems and customizable syllabus and learning path. Also, integrating these features will provide better personalized and customizable contents to learners along with the ability to choose a learning path that best suits them, that will maximize the learning outcome [5]. Although, there are already a lot of tools that allow sharing and using LOs, but with difficulties that allow their creation. A three-component based Multimedia Learning Object (MLO) framework was proposed that exceeds the limits of other available MLO systems, while it is complied with SCORM standard. The proposed framework consists of three components that are intended for two types of users: authors and learners. The authors have shown that with the first component which is the Media Analysis and Processing Unit (MAPU) through five modules (Video Segmentation, Quality Evaluation and Estimation, Meta-data Extraction, Video Indexing and MLO Structuring) it can create LOs from the received instructional video as input [6]. As is well known, student dropout is a very big and everyday problem we face in the education system. This problem is also very pronounced in MOOCs, in the sense where students / learners leave courses due to various factors, and the percentage of interest becomes lower and lower. There are two types of factors that affect this problem and they are: Student related (lack of motivation, lack of time, insufficient background knowledge and skills) and MOOC related (course design, isolation and lack of interactivity, hidden costs). The current state-of-the-art approaches dealing with MOOC dropout prediction are mostly using clickstream features as engagement patterns. There are plenty of examples where K-Means, Decision Trees, Deep Neural Network (DNN) and other machine learning techniques have been used [7]. There are many challenges in using machine learning techniques regarding student dropouts problems, such as: lack of sufficient sample data, managing large masses of unstructured data, data variance, high data imbalance, availability of publicly accessible dataset, lack of standard for creating and representing clickstream data, and student schedule related challenges [7]. The are presented various recommendations and proposals towards useful and effective predictive solutions for dropout predictions, which may not only assist in developing generalizable solutions across different MOOCs but also help lecturers to timely intervene if they foresee dropouts during a course [7]. As e-Learning platforms are becoming more accessible, where their main goal is to provide a smarter way of learning. The new paradigm of e-Learning is also known as Cloud e-Learning where the whole process is done through Cloud services, where it allows a much easier and more flexible way. Part of these platforms are recommendation systems that try to recommend learners courses or materials that are similar to their learning path. There is a recommendation system based on hierarchical clustering or the so-called Cloud e-Learning Recommendation System (CeLRS) [8]. The whole CeLRS process contains several steps (Information Retrieval, Text Mining and Mapping Process). In Mapping Process CeL Learning Objects (CeLLOs) the relevant CeLLOs are generated in the same cluster which are categorized hierarchically, partial, graph-based, etc. via various clustering algorithms. Related study in this field attempts to recommend a hierarchical approach for clustering the CeLLOs as part of topics and sub-topics of computer domain based on a specific ontology and a vector space model [8]. Also, previous research has tried to contribute to the classification systems in pedagogical content, where a rather large focus has been on the content classification of video lectures.In a previous study [9] 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 a combination of support vector machines (SVM) and optical character recognition (OCR) to classify visual content into figures, text and equations [9]. The assumptions that they have made is that there is no clear demarcation between handwritten text lines and figures. As the handwritten text varies significantly in size and there is also a lack of uniform edge density due to chalk, so the traditional OCR techniques are not effective in this context [9]. Furthermore, classifying lecture content into figure, text and equation can be useful for applications like: automatic structuring and indexing of lecture video, creating multimedia learning objects for e-learning and for useful meta-data extraction [9]. Also, other research in this field has presented the classification and organization of pedagogical documents using domain ontology [10].

3.3 Phases Overview

The general overview of the classification model phases as described above.

I. **Feature Extraction**- Since we now have more knowledge about the fact that the data in most cases is not structured, and as such there must be stages where they will be better constructed. This phase or process in which an attempt is made to remove and clean a piece of text from metadata and characters or letters that are not useful in further processing, where they are often referred to as noisy data. Through this, we will convert one piece of text or document into a so-called *structured feature space*, which will be useful to us when using a classifier. If data cleaning and removal of unnecessary characters or letters are not applied, it can directly affect the performance of the system to lead to adverse and inaccurate results.

   We can observe that one of the theories is that the text can be presented in two ways as [11]:

   - ***bag-of-words*** - in this method the text is divided into a set of words where a number is also indicated which shows how many times the word was found in the text.
   - ***strings format*** - each sequence of words in the text is displayed in this way.

   We do not want to describe every technique or method that makes up this phase, because there are a large number of them that are covered and perhaps even more that exist that are not covered in this paper. The idea is to show only a few of them that we will use during further work. The various methods used during this phase to clear the data and prepare it for further processing are [3]:

- ***Tokenization*** - is the process of separating a piece of text into smaller units called tokens. The way the token is formed is based on a delimiter, which in most cases is space. Also, tokens can be words or subwords, but also at a lower level based on characters.
- ***Stop Words*** - are words that are commonly used in one language, that are not needed in the data processing part, and in most cases are ignored because they take up more space in the database, and affect longer processing times. In English stop words are words like: "a", "the", "an", "it", "in", "because", "what", and many others.
- ***Capitalization*** - is the part where it is necessary to identify the correct capitalization of the word, where the first 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, punctuations, 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 the word loses its meaning. This problem can be solved in two ways: with edit distance and another with overlap using k-gram [20].
- ***Stemming*** - is a process where more morphological variants are produced than the base word or the so-called root 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 the text.

- ***Syntactic N-Gram***
  - Weighted Words (such as TF and TF*IDF)
  - Word Embedding (such as Word2Vec, GloVe, FastText)

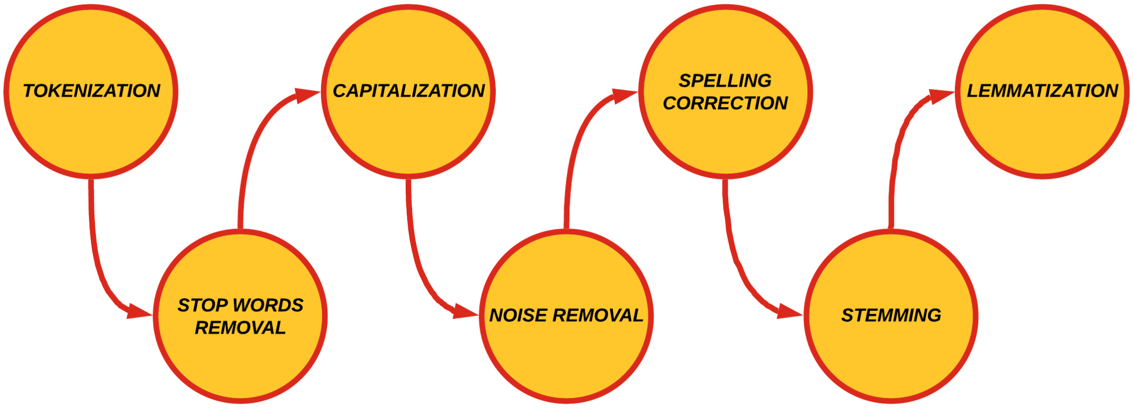


Figure 3. Techniques of data preprocessing phase.

II. **Dimension Reductions** - As we can conclude from the name itself that in this step the goal is to transform from a high-dimensional space to a low-dimensional space. The reason for this is that we strive to improve performance, speed uptime, and reduce memory complexity. There are many types of algorithms or techniques in this step such as:

- ***Principal Component Analysis (PCA)*** - is the most widely used unsupervised technique for dimensionality reduction of large datasets in a interpretable way. This method works on the principle of finding as many variations as possible, where with the help of creating new variables that serve as linear functions of data, whose variation is maximized. By finding these variables, Principal Components (PCs) solve the eigenvalue or eigenvector problem [16].

- ***Non-negative Matrix Factorization (NMF)*** - is a group of algorithms that one matrix is factorized into two matrices that contain non-negative elements. To better illustrate this method, we can explain via the formula where one matrix X is factorized into two matrices W and H.

$$\boldsymbol{X} \approx \boldsymbol{W} \ \boldsymbol{H}[17]$$

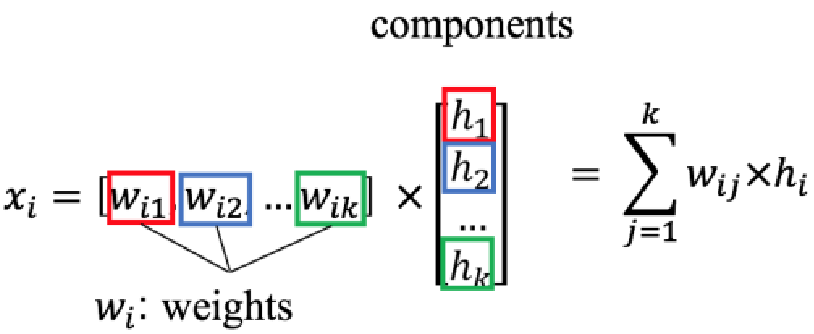
Each of these matrices consists of a specific number of rows. We consider that the matrix X, W and H consist of k rows.

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_k \end{bmatrix} \quad W = \begin{bmatrix} w_1 \\ w_2 \\ \dots \\ w_k \end{bmatrix} \quad H = \begin{bmatrix} h_1 \\ h_2 \\ \dots \\ h_k \end{bmatrix}$$

[17]

Figure 4. Factorization of matrices.

As we have already mentioned, the elements of the matrix X are $x_1, x_2, \dots x_k$. The matrices W are$w_1, w_2, \dots w_k$. And the matrices H are$h_1, h_2, \dots h_k$. Using this we can visualize our equation by taking each i-th row in the X matrix.


components
$x_i = [w_{i1}\ w_{i2} \dots w_{ik}] \times \begin{bmatrix} h_1 \\ h_2 \\ \dots \\ h_k \end{bmatrix} = \sum_{j=1}^{k} w_{ij} \times h_i$
w_i: weights

[17]

Figure 5. Equation of weights and components of matrices.

In Figure 5 we can see that x_i is the weighted sum of some components, so that each row of the W matrix represents the weight of the component that we consider each row in the H matrix. NMF is used in most cases in facial analysis and topic modeling, and is implemented in the scikit-learn library which helps us to use this method more easily.

- ***Linear Discriminant Analysis (LDA)*** - is one of the supervised dimensionality reduction methods, which attempts to transform features into lower dimensional space by maximizing the ratio between class variants to the within-class variance [18].

$$J(W) = \frac{W^T S_B\, W}{W^T S_W\, W}$$

[18]

Figure 6. Linear Discriminant Analysis equation.

There are two types of LDA techniques that deal with classes, and they are [18]:

1. *Class-dependent* - separate lower dimensional space is calculated for each class.

2. *Class-independent* - each class is considered as a separate class in the relation with other classes.

The main goal of the LDA technique is to project the original data into the matrix but into the lower dimensional space, and this process in most cases consists of three steps:

Step 1. In this step, the distance between the means of different classes is calculated, which is called between-class variance or between class matrices [18].
Step 2. In this step, the distance between the mean and the samples of each class, which is called within-class variance or within-class matrices [18].
Step 3. In the last step, the idea is to construct the lower dimensional space in which will be achieved maximization of the between-class variance and minimization of the within-class variance.

- ***Kernel PCA*** - as we already know that traditional PCA allows only linear dimensionality reduction, but in cases where the data are more complex, traditional PCA becomes helpless. And in this sense, the PCA kernel allows us to generalize traditional PCA to nonlinear dimensionality reduction [19]. Kernel PCA is quite similar to Support Vector Machines (SVM), where it uses kernel functions to project datasets into a higher dimensional feature space, where they are linearly separable.

We can see that it pays more to use dimensional reduction for pre-processing than some kind of classification algorithm [3].

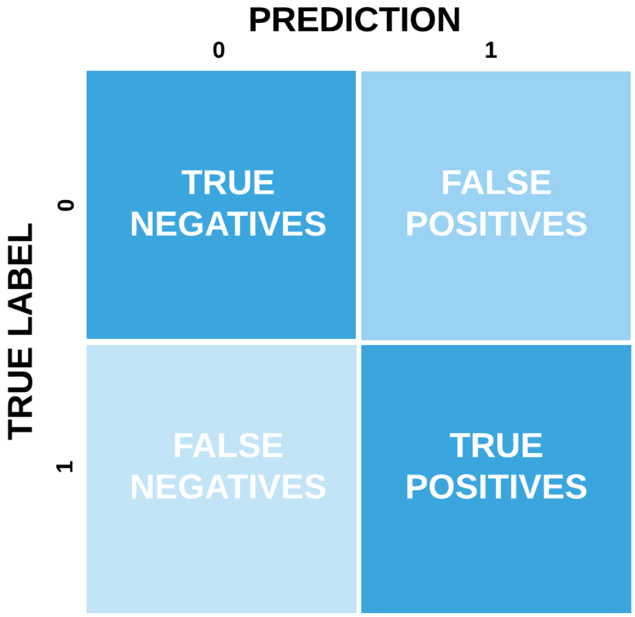
III. **Classifier Selection** - One of the main concerns is to choose the right classifier model that will be able to perform with a certain set of data to achieve the desired results. Choosing the right classifier model is not an easy task, and is a challenge that is also referred to in the literature 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 complementarity [12]. 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. One of the first models to deal with ASP is Rice's model, which recommends several features and options to be controlled when selecting an algorithm [13].

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)
- Combination Techniques

IV. **Evaluations -** One of the most important steps that text mining systems consist of is the Evaluation part. In this step, algorithms are analyzed or scored to assess how efficiently they performed. One of the problems where evaluating just about every method or algorithm is not possible, but only a couple of them and that is the reason for the lack of data and standard evaluation methods. It should also be suggested that comparing different parameters or metrics with this method is not an easy task.

There is a so-called ***confusion matrix*** table in which classification metrics such as *True Positives (TPs)*, *False Positives (FPs)*, *False Negatives (FNs)* and *True Negatives (TNs)* are calculated and presented [15].



TRUE LABEL \ PREDICTION	0	1
0	TRUE NEGATIVES	FALSE POSITIVES
1	FALSE NEGATIVES	TRUE POSITIVES

Figure 7. Confusion Matrix table.

Figure 7 shows a confusion matrix table in which the prediction results are displayed horizontally, while a label that is positive or negative is shown vertically.

3.4 Classification Algorithms

This section will describe the K-Nearest Neighbours (KNN) and Recurrent Neural Networks (RNN) algorithms that will be used further in our experiment.

3.4.1 K-Nearest Neighbours (KNN)

K-Nearest Neighbors (KNN) is one of the techniques that can be 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 [21].

In order to identify which of the K instances in the dataset is similar to the received input variable, in this case we measure the distance. One of the most popular distance measures used is the Euclidean distance, which is shown by the formula:

$$\text{Euclidean Distance (a, b)} = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2}[21]$$

Figure 8. Euclidean Distance between two points.

As we can see, the Euclidean Distance is calculated by the formula as the square root of the sum of the differences between two points a and b over all the input attributes i. The greater the distance between two points in the KNN model, the smaller the similarity between them.

There are several assumptions regarding the KNN model that should be considered:

- KNN is a non-parametric algorithm, where there are no assumptions regarding the data before the model is used.
- As we already know, in most other algorithms the data are divided into test sets and training sets, while in this model they are not. This model does not generalize data, but takes the entire dataset.
- There is no need to learn the model, the whole job happens at the time of prediction, and this is known as the Lazy Learning concept.

There are a number of steps that the KNN algorithm goes through, and these steps are [22]:

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.

One of the main problems with the KNN algorithm is to determine the exact number of K neighbors, and if this number is not determined correctly it can lead to incorrect prediction results.

To conclude, KNN is one of the very efficient algorithms used for both classification and regression, and is very simple and easy to use. KNN does not make any assumptions about the data, and can be used for a variety of problems. The disadvantage of the KNN algorithm is memory load and a long process time, because this algorithm uses the entire data set.

3.4.2 Recurrent Neural Networks (RNN)

Recurrent Neural Networks (RNN) are types of artificial neural networks that allow previous outputs to be used as inputs while having hidden states [24]. 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 [23].

Depending on the mapping of inputs from output, there are different topology types for RNNs, and they will be shown in the figures below.

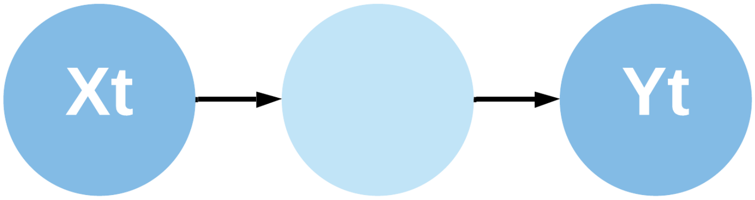


Figure 9. One-to-one type of RNN.

In Figure 9 you can see one-to-one mapping, where $X_t = Y_t = 1$. This type of mapping in RNN is in most cases used in examples as a traditional neural network [24].

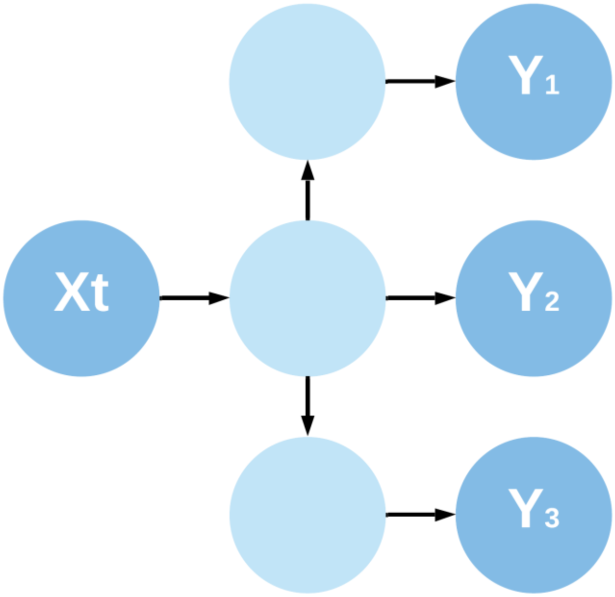


Figure 10. One-to-many type of RNN.

In Figure 10 one-to-many type of mapping is presented, where $X_t = 1;\ Y_t > 1$. This type of mapping is used in music generation applications [24].

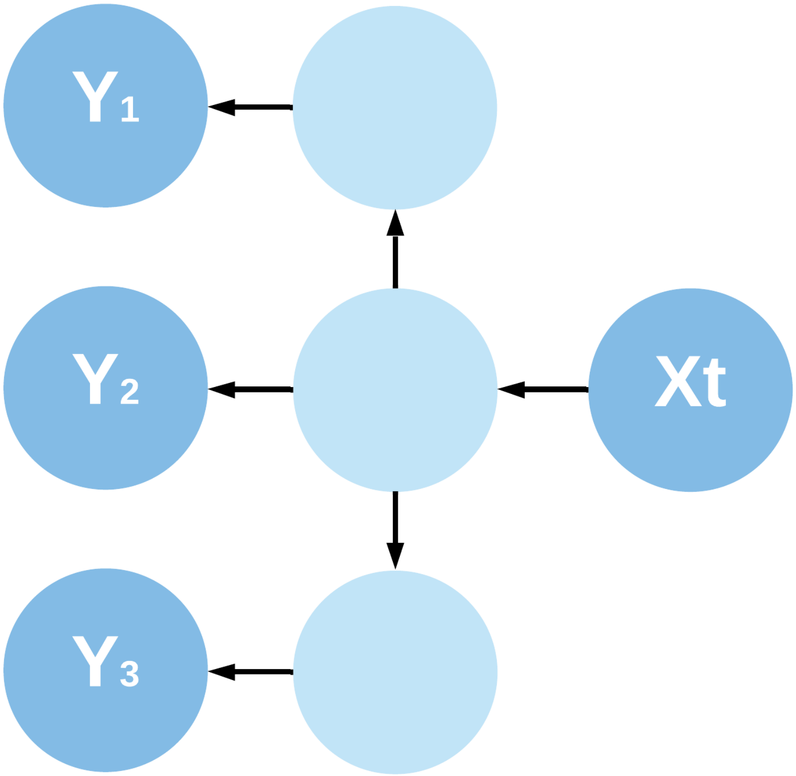


Figure 11. Many-to-one type of RNN.

Figure 11 shows a similar type of mapping as in the previous figure, only this type of mapping is many-to-one compared to the previous one which is one-to-many. For this mapping the presentation is $X_t > 1;\ Y_t = 1$.

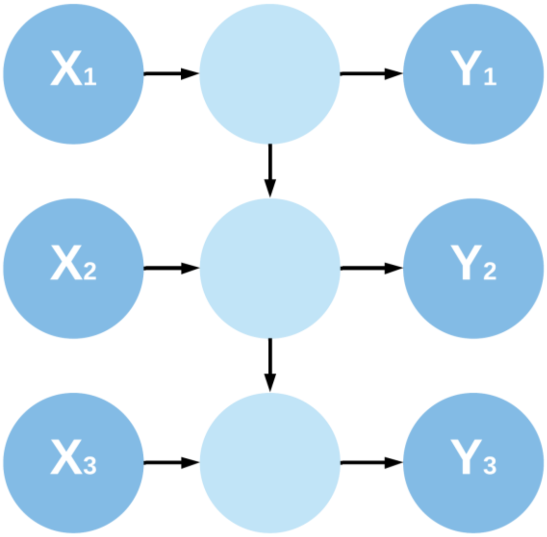


Figure 12. First example of many-to-many type of RNN.

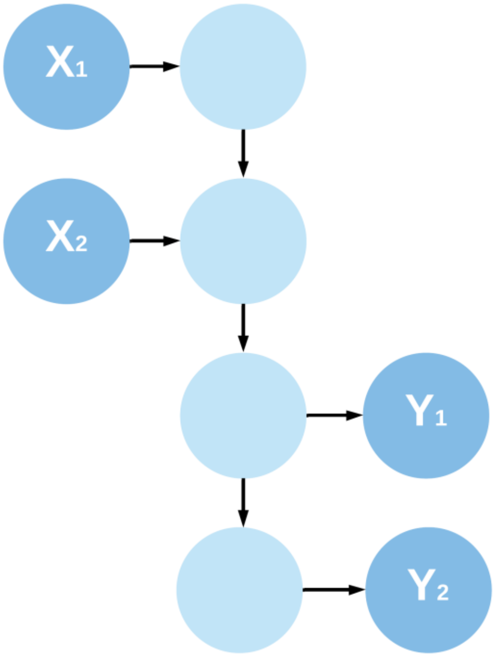
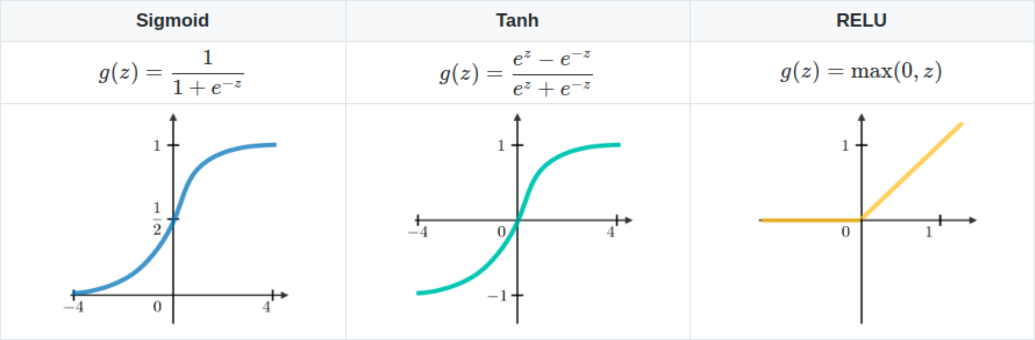


Figure 13. Second example of many-to-many type of RNN.

Figure 12 shows the type of RNN mapping where $X_t = Y_t$, while Figure 13 shows when $X_t \neq Y_t$,. Both of these figures represent a many-to-many type of mapping, except that mapping in both cases is not direct, but in one example is indirect via hidden states.

In the neural network implementation process, it is very important to decide which activation function to use in the hidden and output layer, with the aim of enabling back-propagation to update weights and biases. We will show the three most common activation functions that are Sigmoid, Tanh, Relu, and they are shown below:



Sigmoid	Tanh	RELU
$g(z) = \frac{1}{1+e^{-z}}$	$g(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$	$g(z) = \max(0, z)$

1
$\frac{1}{2}$
−4
0
4
1
−4
0
4
−1
1
0
1

[24]

Figure 14. Representation of activation functions.

3.5 Application and Domains

As stated earlier in the paper, the use of text classification methods is found in various types of applications in several different domains. There are several reasons why the application is so extensive, first one of the reasons is the existence of different classification techniques available today. Another reason is that businesses and companies have direct profits for various purposes after one of the techniques is applied and results are obtained.

Text classifications are used in different domains, where some of the examples are:

- ***Risk Management and Analysis*** - text mining is very widely used especially in the financial industry where risk analysis is much needed to increase the ability to mitigate risk, and through this, a large number of documents are managed and analyzed.

- ***News Filtering*** - as is the case today with most media and portals in an electronic edition, where all their information is contained in digital documents. Over time, it becomes quite a problem to more easily search or manage this data, where it is necessary to use text mining techniques to achieve a better organization and structure of documents [11].

- ***Document organization and retrieval*** - here it can be shown how the use of supervised methods can lead to better classification in large bookstores, scientific collections, bookstores, social sites, etc. The goal is to better structure this data hierarchically [11].

- ***Opinion Mining*** - in this domain it is hinted that customer reviews and opinions are very important and they are often recorded in the form of a text document, where interesting data and statistics that would be needed for various purposes of improving products or services are processed and extracted.

- ***Email Classification and Spam Filtering*** - to separate the emails that are sent daily in large numbers between billions of users around the world, from those emails or so-called spams that try to take personal information from email users, and use them for bad purposes. This process is automatically advanced using text mining, where emails, spams, and junk emails are filtered and classified, and these applications are called email filters or spam filters [11].

- ***Business Intelligence*** - in this domain, text mining is used to make it easier to make decisions and conclusions that application users could use more effectively.

Through this one can analyze and decide regarding the information collected from the client or user [14].

- ***Data Analysis in Social Media*** - text mining techniques helps to analyze data on social networks, such as the number of likes, comments, shares of posts, users' interests for different pages or groups, and their activity on social networks [14].

4. METHODOLOGY

In this chapter, we will focus on the methodology of work we used during the research part, as well as designing and implementing an experiment part.

First of all, we tried to review the existing literature and scientific papers in order to get better acquainted with the research that was done earlier in this domain, and what other authors tried to contribute in order to gain new knowledge when it comes to review and research challenges in classification of documents in a pedagogical content. After this, we gained a lot of information about document classification, where several authors tried using different techniques to get better results, and helped us a lot to have a better view when selecting the classification algorithm when designing our experiment.

During the literature review, we noticed that there are different models that contain certain flow processes or phases, and that the idea of these models is the same, but differs in the order and design of these phases. We have chosen in our paper to use the four-phase model that most text classification systems use. As an input to this model we provide a dataset that we have chosen for the further process. This model consists of four phases and they are: Feature Extraction, Dimension Reductions, Classifier Selection and Evaluation.

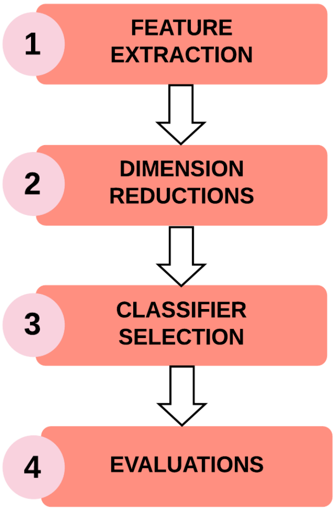


Figure 15. Representation of four-phase experimental model.

The research method used in this paper is a secondary data collection based on provided datasets. After a long research and analysis of several datasets, and various questions that we can further explore, during further work on this paper we use a dataset that was manually collected from the Coursera platform, and which contains four attributes. The first three attributes are the categories of the video lesson and are divided into three levels: General level, Specific level and Course level, and also the fourth attribute is the textual transcript taken from the video lesson. This dataset is provided by Professor Ali Shariq Imran, where it was used in one of his scientific papers [25]. Our further work in this paper will be based on review dataset, data preprocessing, selection of two classifiers, creation of models for both classifiers, comparison of results from previously created models, creation of test scenarios and evaluation of models.

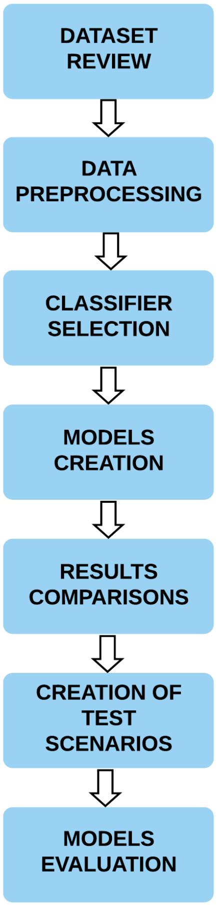


Figure 16. Organization and structure of processes.

5. DESIGN AND IMPLEMENTATION OF EXPERIMENT

In this chapter, we will focus on explaining the experimental part of the paper. The structure of this chapter will be divided into sections where the data collection process will be explained first, followed by a section showing how we prepared and cleaned our data for the further process. And after that, in the last section, the selected models, their architecture and implementation during the experiment will be presented.

5.1 Data Collection

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 can be used to achieve the desired results. We analyzed several open-source datasets collected from multiple MOOCs platforms, which were published on the Kaggle. One of these datasets consisted of online course data from the MITx and HarvardX, where we came up with the idea of classifying the number of certified participants based on the course subject category who completed more than 50% of the course content. One of the reasons we did not want to analyze and use this dataset further in our experiment was due to the insufficient number of records, there were a total of 291 records, which did not meet the requirements of this research. Thanks to Professor Ali Shariq Imran who expressed a desire to help us, and shared with us the dataset he used to conduct the experiments and validate the proposed video classification framework that uses the transcript from the video as feature representations [25]. 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. To conduct our experiment we will further use this dataset, where we will try to classify the course transcript based on the course level or also known as the fine-grained level to which it belongs.

5.2 Data Preparation and Preprocessing

In order for the data to be in the correct format for further analysis and modeling process, the data needs to be prepared, cleaned, and transformed. This is one of the very important steps to primarily improve the quality of the data, and thus the results of the learned models, because the data is directly fed into the model. 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 word '[MUSIC]' which was in most of the transcript records. After that, we converted the entire textual content of the transcript to lowercase, and removed the non-letters characters. Also, we removed stopwords from the transcript where it helped us reduce the number of features, and kept the model of the appropriate size. In the last part we applied stemming, where we separated their root form from words, where in most cases this process can help improve the accuracy of classification, and keep the vocabulary in more standardized format.

5.3 Classification

In this part, the creation of a model with the aim of classifying transcripts depending on the category of course level will be presented. As we have already stated in the paper objectives, we selected two appropriate multi-class classification techniques, and in this case we chose to create the first model with K-Nearest Neighbors and the second model with Recurrent Neural Networks. Both of these techniques are well known in solving classification problems. RNN is one of the most well-known Deep Learning methodologies, while KNN is one of the simplest and easy to implement machine learning algorithms. The implementation for both of these techniques used in the experiment has been explained in Chapter 3.

6. RESULTS AND DISCUSSION

In this chapter, we will present the results and conclusions on the results obtained from the classification of textual content (transcripts) of a video lesson based on the category to which they belong. In the previous chapter we explained the designs and implementation of the experiment, while in this chapter the focus will be on discussing and comparing the results. We will first show the results we obtained with the K-Nearest Neighbors algorithm, for all three categorization levels General level, Specific Level and Course Level, depending on the number of categories they contain. We will then repeat the same process for the results obtained with Recurrent Neural Networks.

To evaluate the performance of our models and algorithms, we used evaluation metrics: ***precision***, ***recall, f1 score***, and ***accuracy***. In order to have a better idea regarding these evaluation metrics, we will describe each of them below.

Precision - represents the evaluation metrics that points out how accurate the model is based on those which are positively predicted. This evaluation metric is really precise when the number of False Positives is high. The formula for calculating the precision metric is as follows:

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

Recall - this evaluation metric is kind the same as Precision, but in comparison with it, this method is really precise when the number of False Negatives is high. The formula for calculating the recall metric is as follows:

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

F1 Score - is an evaluation metric that is used when you want to find a balance between two mentioned metrics above - Recall and Precision. The formula for calculating the f1 score is as follows:

$$\text{F1 Score} = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Accuracy or classification accuracy - is an evaluation metric that represents a ratio between the number of correct predictions to the total number of predictions that are made. The formula for calculating the accuracy is as follows:

$$\text{Accuracy} = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}$$

Table 1 shows the classification results with the K-Nearest Neighbours algorithm. The table is structured so that the columns show the corresponding values of the evaluation metrics, and the rows show the categorization of a particular level. We will explain the analysis of the obtained results for each level individually. As we can see from the table, the general level based on the precision metric has shown a very good result of 92.63% accuracy. For a specific level, we can observe that 87.89% accuracy is estimated by precision metrics. And at the course level, also based on the precision metric, we can see the achieved result of 78.59% accuracy. If we analyze the obtained results for all three levels, we notice that the percentage of accuracy decreases, where the highest accuracy is achieved at the general level, followed by a specific level, while the lowest accuracy is achieved at the course level. In our case, taking into account the number of categories for a single level by which the video is classified on the Coursera platform differs drastically, where the general level consists of 8 categories, the specific level of 40 categories, and the course level of in the total of 200 categories. With this, we finished analyzing presenting the results with the KNN as a classifier.

Table 1. Classification results with K-Nearest Neighbours.

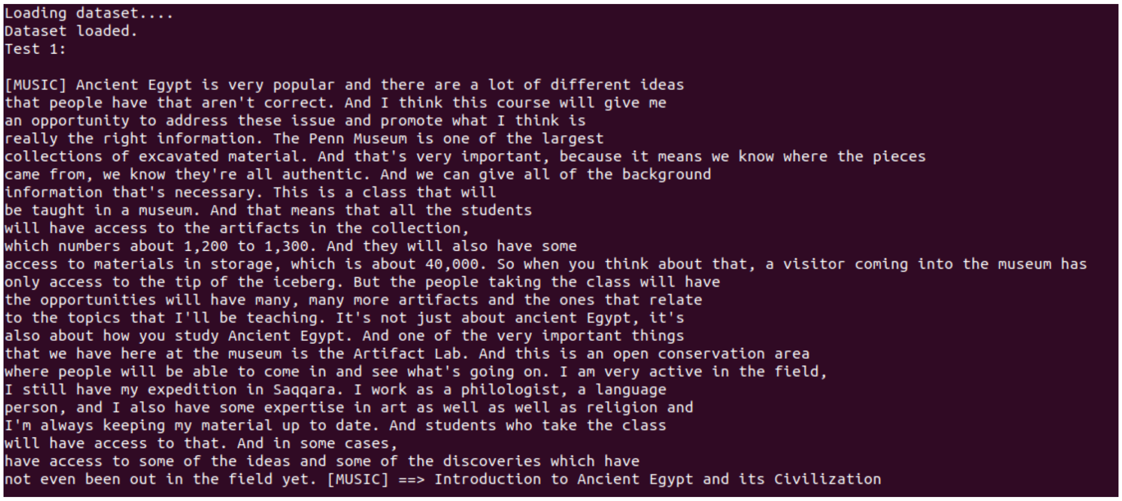
Category	Precision (%)	Recall (%)	F1 Score (%)	Accuracy (%)
General Level	92.63	92.52	92.53	92.52
Specific Level	87.89	87.58	87.49	87.58
Course Level	78.59	76.73	76.11	76.73

Table 2 shows the classification results with the Recurrent Neural Networks, more specifically with an Long Short-Term Memory (LSTM) type. The table is constructed with the same structure as the previous one, where columns show the corresponding values of the evaluation metrics, and the rows show the categorization of a particular level. As we can see from the table, the general level based on the precision metric has shown a very good result of 88.22% accuracy. For a specific level, we can observe that 72.31% accuracy is estimated by precision metrics. And at the course level, also based on the precision metric, we can see the achieved result of 59.49% accuracy. If we analyze the obtained results for all three levels, we notice that the percentage of accuracy decreases, where the highest accuracy is achieved at the general level, followed by a specific level, while the lowest accuracy is achieved at the course level. With this, we finished analyzing presenting the results with the LSTM as a classifier.

Table 2. 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

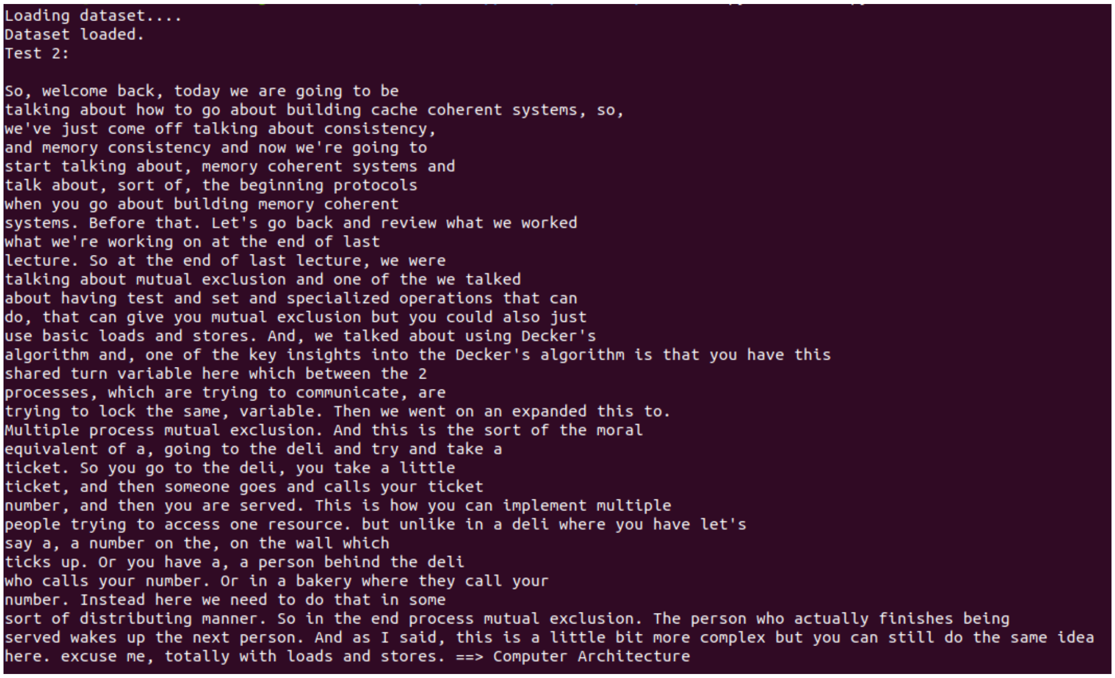
As stated in the objectives of this paper, below we will present several test scenarios, in which we will give as input the transcript we have chosen randomly, and we expect that after classification the model predicts the appropriate course level category in which it belongs.Test scenarios are shown and described below.



```
Loading dataset....
Dataset loaded.
Test 1:

[MUSIC] Ancient Egypt is very popular and there are a lot of different ideas
that people have that aren't correct. And I think this course will give me
an opportunity to address these issue and promote what I think is
really the right information. The Penn Museum is one of the largest
collections of excavated material. And that's very important, because it means we know where the pieces
came from, we know they're all authentic. And we can give all of the background
information that's necessary. This is a class that will
be taught in a museum. And that means that all the students
will have access to the artifacts in the collection,
which numbers about 1,200 to 1,300. And they will also have some
access to materials in storage, which is about 40,000. So when you think about that, a visitor coming into the museum has
only access to the tip of the iceberg. But the people taking the class will have
the opportunities will have many, many more artifacts and the ones that relate
to the topics that I'll be teaching. It's not just about ancient Egypt, it's
also about how you study Ancient Egypt. And one of the very important things
that we have here at the museum is the Artifact Lab. And this is an open conservation area
where people will be able to come in and see what's going on. I am very active in the field,
I still have my expedition in Saqqara. I work as a philologist, a language
person, and I also have some expertise in art as well as well as religion and
I'm always keeping my material up to date. And students who take the class
will have access to that. And in some cases,
have access to some of the ideas and some of the discoveries which have
not even been out in the field yet. [MUSIC] ==> Introduction to Ancient Egypt and its Civilization
```

Figure 17. First test scenario.

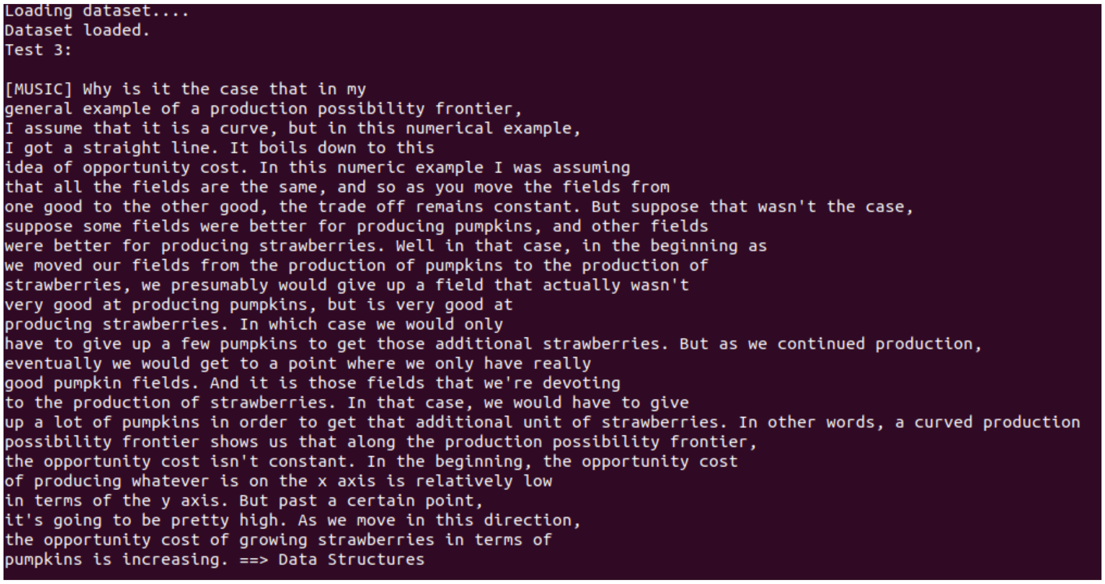


```
Loading dataset....
Dataset loaded.
Test 2:

So, welcome back, today we are going to be
talking about how to go about building cache coherent systems, so,
we've just come off talking about consistency,
and memory consistency and now we're going to
start talking about, memory coherent systems and
talk about, sort of, the beginning protocols
when you go about building memory coherent
systems. Before that. Let's go back and review what we worked
what we're working on at the end of last
lecture. So at the end of last lecture, we were
talking about mutual exclusion and one of the we talked
about having test and set and specialized operations that can
do, that can give you mutual exclusion but you could also just
use basic loads and stores. And, we talked about using Decker's
algorithm and, one of the key insights into the Decker's algorithm is that you have this
shared turn variable here which between the 2
processes, which are trying to communicate, are
trying to lock the same, variable. Then we went on an expanded this to.
Multiple process mutual exclusion. And this is the sort of the moral
equivalent of a, going to the deli and try and take a
ticket. So you go to the deli, you take a little
ticket, and then someone goes and calls your ticket
number, and then you are served. This is how you can implement multiple
people trying to access one resource. but unlike in a deli where you have let's
say a, a number on the, on the wall which
ticks up. Or you have a, a person behind the deli
who calls your number. Or in a bakery where they call your
number. Instead here we need to do that in some
sort of distributing manner. So in the end process mutual exclusion. The person who actually finishes being
served wakes up the next person. And as I said, this is a little bit more complex but you can still do the same idea
here. excuse me, totally with loads and stores. ==> Computer Architecture
```

Figure 18. Second test scenario.

Figure 17 shows the first test scenario where our model predicted the *"Introduction to Ancient Egypt and its Civilization"* course level category for the entered transcript. While Figure 18 envisages a *“Computer Architecture”* course level category. We can see that both of these results are correct because these two transcripts belong to these categories, where we can also manually check in the dataset.



```
Loading dataset....
Dataset loaded.
Test 3:

[MUSIC] Why is it the case that in my
general example of a production possibility frontier,
I assume that it is a curve, but in this numerical example,
I got a straight line. It boils down to this
idea of opportunity cost. In this numeric example I was assuming
that all the fields are the same, and so as you move the fields from
one good to the other good, the trade off remains constant. But suppose that wasn't the case,
suppose some fields were better for producing pumpkins, and other fields
were better for producing strawberries. Well in that case, in the beginning as
we moved our fields from the production of pumpkins to the production of
strawberries, we presumably would give up a field that actually wasn't
very good at producing pumpkins, but is very good at
producing strawberries. In which case we would only
have to give up a few pumpkins to get those additional strawberries. But as we continued production,
eventually we would get to a point where we only have really
good pumpkin fields. And it is those fields that we're devoting
to the production of strawberries. In that case, we would have to give
up a lot of pumpkins in order to get that additional unit of strawberries. In other words, a curved production
possibility frontier shows us that along the production possibility frontier,
the opportunity cost isn't constant. In the beginning, the opportunity cost
of producing whatever is on the x axis is relatively low
in terms of the y axis. But past a certain point,
it's going to be pretty high. As we move in this direction,
the opportunity cost of growing strawberries in terms of
pumpkins is increasing. ==> Data Structures
```

Figure 19. Third test scenario.

We can see in Figure 19 that the entered transcript does not belong to the category *"Data Structures"*, which our model predicted. In fact, this transcript belongs to the category *"Microeconomics: The Power of Markets"*, and in this scenario, our model gave us the wrong result.

7. CONCLUSION

So far, we have presented and discussed the classification results of the experiment we conducted within this paper for all three category levels both with KNN and LSTM. We can conclude that better results were 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 can conclude that the classification results are directly affected by the number of categories that each level contains. From results above we can clearly see that KNN in most cases performed much better than LSTM which can be best noticed at Course level category. This fact depends on several factors. First, is the quantity of data required for LSTM, and this is because a large number of categories increases the complexity of the problem, and thus requires more data to train the model. Another reason why LSTM has not given higher accuracy is due to the high similarity of different transcripts. Many of the transcripts belonging to different classes at the third level had many similarities in the context of the sentences and keywords, so the model could not properly distinguish in which class the transcripts belonged.

In closing, this research can be improved by investigating more on recurrent neural networks like, applying hyperparameters tuning. Another step that can be taken is using other classification techniques such as Support Vector Machines, Random Forests and Decision Trees. These and other potential improvements are left to be addressed in future work.

8. REFERENCES

[1] F. Sebastiani, "Machine learning in automated text categorization," *ACM Computing Surveys*, vol. 34, no. 1, pp. 1–47, 2002.

[2] Samta Tembhekar, Monika Kanojiya. A Survey Paper on Approaches of Natural Language Processing (NLP), International Journal of Advance Research, Ideas and Innovations in Technology, www.IJARIIT.com.

[3] K. Kowsari, K. J. Meimandi, M. Heidarysafa, S. Mendu, L. E. Barnes, and D. E. Brown, "Text Classification Algorithms: A Survey," 2019.

[4]F. Dalipi, S. Y. Yayilgan, A. S. Imran, and Z. Kastrati, "Towards understanding the MOOC trend: pedagogical challenges and business opportunities," *in International Conference on Learning and Collaboration Technologies*, pp. 281–291, Springer, 2016.

[5]K. Pireva, A. S. Imran, and F. Dalipi, "User behaviour analysis on LMS and MOOC," in *2015 IEEE Conference on e-Learning, e-Management and e-Services (IC3e)*, pp. 21–26, IEEE, 2015.

[6]A. S. Imran and F. A.Cheikh, "Multimedia Learning Objects Framework for E-Learning," in *2012The International Conference on E-Learning and E-Technologies in Education*, IEEE, 2012.

[7]F. Dalipi, A. S. Imran, and Z. Kastrati, "MOOC dropout prediction using machine learning techniques: Review and research challenges," in *2018 IEEE Global Engineering Education Conference (EDUCON)*, pp. 1007–1014, IEEE, 2018.

[8]K. Pireva and P. Kefalas, "A Recommender System Based on Hierarchical Clustering for Cloud e-Learning," in *2017 11th International Symposium on Intelligent Distributed Computing*, IEEE, 2017.

[9]A. S. Imran and F. Alaya Cheikh, "Blackboard content classification for lecture videos," in *2011 18th IEEE International Conference*, pp. 2989-2992, Image Processing (ICP), 2011.

[10]A. S.Imran and Z. Kastrati, "Pedagogical Document Classification and Organization Using Domain Ontology," in *2016 Lecture Notes in Computer Science 9573*, 2016.

[11] C. C. Aggarwal and C. Zhai, “A Survey of Text Classification Algorithms,” *Mining Text Data*, pp. 163–222, 2012.

[12] I. Khan, X. Zhang, M. Rehman, and R. Ali, “A Literature Survey and Empirical Study of Meta-Learning for Classifier Selection,” *IEEE Access*, vol. 8, pp. 10262–10281, 2020.

[13] J. R. Rice, “The Algorithm Selection Problem,” *Advances in Computers Advances in Computers Volume 15*, pp. 65–118, 1976.

[14] R. Talib, M. Kashif, S. Ayesha, and F. Fatima, “Text Mining: Techniques, Applications and Issues,” *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 11, 2016.

[15] J. Lever, M. Krzywinski, and N. Altman, “Erratum: Corrigendum: Classification evaluation,” *Nature Methods*, vol. 13, no. 10, pp. 890–890, 2016.

[16]"Principal component analysis: a review and recent developments | Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences", *Royalsocietypublishing.org*, 2020. [Online]. Available: https://royalsocietypublishing.org/doi/10.1098/rsta.2015.0202#d3e289. [Accessed: 22- Nov- 2020].

[17]"A Practical Introduction to NMF (nonnegative matrix factorization)", *Machine Learning Explained*, 2020. [Online]. Available: https://mlexplained.com/2017/12/28/a-practical-introduction-to-nmf-nonnegative-matrix-factorization/. [Accessed: 22- Nov- 2020].

[18] A. Tharwat, "Linear Discriminant Analysis: An Overview", 2015.

[19]Q. Wang, "Kernel principal component analysis and its applications in face recognition and active shape models", *arXiv preprint arXiv:1207.3538*, 2012.

[20]"Spelling correction", *Nlp.stanford.edu*, 2020. [Online]. Available: https://nlp.stanford.edu/IR-book/html/htmledition/spelling-correction-1.html. [Accessed: 22- Nov- 2020].

[21]J. Brownlee, *Master Machine Learning Algorithms*. 2016.

[22]D. Nelson, "What is a KNN (K-Nearest Neighbors)?", *Unite.AI*, 2020. [Online]. Available: https://www.unite.ai/what-is-k-nearest-neighbors/. [Accessed: 22- Nov- 2020].

[23]"What are Recurrent Neural Networks?", *Ibm.com*, 2020. [Online]. Available: https://www.ibm.com/cloud/learn/recurrent-neural-networks. [Accessed: 22- Nov- 2020].

[24]"CS 230 - Recurrent Neural Networks Cheatsheet", *Stanford.edu*, 2020. [Online]. Available: https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks. [Accessed: 22- Nov- 2020].

[25] Z. Kastrati, A. S. Imran, and A. Kurti, “Integrating word embeddings and document topics with deep learning in a video classification framework”, *Pattern Recognition Letters*, vol. 128, pp. 85-92, 2019.