Creating Video Sequence Alternatives Across MOOCs Based on Document Similarity
Creating Video Sequence Alternatives Across MOOCs Based on Document Similarity

By

Alexander Grooff
4406079

in partial fulfilment of the requirements for the degree of

Master of Science
in Computer Science / Software Technology

at the Delft University of Technology,
to be defended publicly on 14-06-2018.

Supervisor: Prof. Dr. ir. G.J. Houben
Thesis committee: Dr. C. Lofi
M. A. Zúñiga Zamalloa, PhD
WIS, EEMCS, TU Delft
WIS, EEMCS, TU Delft
ES, EEMCS, TU Delft

This thesis is confidential and cannot be made public until 14-06-2018.

An electronic version of this thesis is available at http://repository.tudelft.nl/.
## Contents

1 Introduction 7  
  1.1 Research Questions and Contributions 8  

2 Related Work 9  
  2.1 Improving Lecture Content 9  
  2.2 Analysing MOOC Sequence Order Relationships 9  
  2.3 MOOC Content Analysis 9  

3 Theoretical Definitions 10  
  3.1 Schematic Syntax 10  

4 Data Properties 11  
  4.1 MOOC Structure 11  
  4.2 Video Data 11  

5 Pre-study 12  
  5.1 Experiment Setup 12  
  5.2 Results 12  

6 Method 13  

7 Similarity 13  
  7.1 Transcription similarities 13  
    7.1.1 Tf-idf 14  
    7.1.2 Word2Vec 14  
    7.1.3 Doc2Vec 14  
  7.2 Metadata Similarity 14  
    7.2.1 Title similarity 14  
    7.2.2 Difference in Video Duration 14  
  7.3 Combining Ratings 15  

8 Creating Alternative Sequence Orders 15  

9 Evaluation 16  
  9.1 Quality of alternative sequence orders 16  
    9.1.1 Experiment Setup 16  
    9.1.2 Similarity Thresholds 16  
    9.1.3 User Rating Review 17  
    9.1.4 Creating successors based on similarity 18  
  9.2 Optimizing run-time performance 18  

10 Discussion 19  
  10.1 Reflection on research questions 19  
  10.2 User study 19  
  10.3 Cyclic Graph 20  

11 Conclusion 20  

12 Future Work 20  
  12.1 Data Variety 20  
  12.2 User Study 20  

13 Acknowledgements 20
Abstract

Since the introduction of Massive Online Open Courses (MOOCs) in 2008, the number of MOOCs offered by universities has increased enormously. Over 700 universities offer a total of 5924 MOOCs. Each MOOC holds a sequence of 10 to 140 videos and are meant to help online learners understand a given topic. These videos are intended to be watched in a specific, sequential order. This order can be seen as a chain where video 1 provides enough prerequisite knowledge for video 2, video 2 for video 3 and so on. In theory that is fine, however, this order is not necessarily the only ideal order to watch videos in. The online learner might want to take shortcuts through the MOOC’s videos, find additional information on the topic in another video or explore what other subjects are available to study. In other words, the online learner could benefit from improved navigational features to better explore the surrounding relevant videos. By branching out the viewing order towards videos in other MOOCs, we add alternative sequence orders as possible learning paths for the online learner. This can help online learners plan out an alternative sequence order based on their own interests and backgrounds. The chain-like sequence order as described above, where one video provides the prerequisite knowledge for its successor, is crucial for creating alternative sequence orders. When video 1 from MOOC A is found to be similar to video 5 from MOOC B, it can be assumed that video 1 provides the prerequisite knowledge for video 6 in MOOC B. Similar videos between the MOOCs are found by comparing their transcripts, titles and durations. Based on this similarity a new sequence order is set up from one video to its similar counterparts’ successor video. We structure these alternative sequence orders by representing the videos from MOOCs as nodes into a graph and connect them with directed edges that represent the sequence order between videos. Several methods for finding similarities have been compared to find the most accurate way of comparing transcriptions where the method doc2vec yields the most accurate similarity cases. Evaluation based on a user study shows that our method creates alternative sequence orders that have 58.7% positive user ratings. This can be compared to semi-randomly picked sequence orders and the original sequence order defined in the MOOC, which have 8.0% and 76% positive user ratings respectively. This shows that the alternative sequence orders are significantly better than the semi-randomly picked sequence orders, and comparable but not as good as the original sequence order defined in the MOOC.
Creating Video Sequence Alternatives Across MOOCs Based on Document Similarity

Alexander Grooff

1. Introduction

MOOCs provide an easy way to let an online learner study a subject. First seen in 2008 and becoming popular in 2012[6, 13], MOOCs have seen a growing increase of interest over the last few years. The total number of MOOCs has seen a growth rate of 66% in 2016-2017 to a total of 5924 MOOCs[3], with each MOOC containing 10 to 140 videos (based on our dataset gathered from Coursera[1] and Khan Academy[2]). The prescribed video sequential order within a MOOC is made as a one size fits all, while the knowledge and needs when starting a MOOC varies per individual. Knowing that a video within a MOOC is not the only video to cover that subject for all the videos offered in the total amount of MOOCs, it can be that due to the current knowledge and experience an individual starts learning video 1 in MOOC A and then switch to MOOC B, video 7 since that is for that learner with his/her specific background and experience and needs, the most optimal and efficient learning path. The problem is that the large amount of available MOOC videos can overwhelm online learners and can cause them to lose track when trying to deviate from the prescribed sequence, and using multiple MOOCs. The reason to deviate from the sequence can be:

1. The learner only wants to learn a specific part of a MOOC.
2. The learner is under/overqualified for the knowledge presented in a part of the MOOC.
3. The learner wants to explore what other subjects can be studied afterwards other than the subject in the next video.

In all these cases the learner can be helped by an improved way of navigating through MOOCs and its videos, by providing more insights on similarities between videos of different MOOCs so that the online learner can choose what video to watch of what MOOCs and thereby choose the most efficient and optimal path of videos to follow to achieve his/her individual learning goal. This can help online learners to plan their personalized learning path that suits their interests, needs and background.

Finding alternative sequence orders for each video in every MOOC is not an easy task. Doing this manually would require the work of experts, and is not the right solution for two reasons:

1. Having experts looking for alternative sequence orders for all existing MOOC videos would require an impractical amount of effort and time.
2. Even if the alternative sequence orders were to be made by experts, their opinion might be unreliable as they could be susceptible to overestimating the accessibility of their lectures due to their expert blind spot[18].

Instead of doing this manually, this study aims to create a method that finds alternative video sequence orders across MOOCs based on similarities of the content and metadata of the videos. We structure these alternative sequence orders along with the original sequence order described in the MOOC by structuring these videos into a graph, as visualized in figure 1. This graph can be used to provide all possible sequence orders per video to the online learner. For example, this graph can be used to let the online learner figure out their own ideal path to follow, allowing them to find shortcuts between videos or find other subjects to study that interests them.

There have been other studies that look at the sequence order of videos in MOOCs. Most of these studies reach for an opposite goal, as they focus on finding alternative prerequisite knowledge components instead of finding alternative successor videos. This thesis is based on the works of studies that focus on structuring a given topic query of MOOC videos[23], use external sources for finding prerequisite knowledge for lectures[22] or apply a different method than looking for similarities[25, 26].

The method described in this thesis creates an alternative sequence order for a MOOC video by looking across MOOCs for similar videos to that video. Most of these videos have a succeeding video that comes next in the sequence order defined in its MOOC. Since all the similar videos cover the same subject, it means that the successors of these videos can be considered as possible successors to all the similar videos. This makes it so that these videos can have multiple
successors, allowing for alternative sequence orders that can be followed other than the original sequence orders defined in the MOOCs.

These similar videos can be found by comparing data from videos that contain information regarding its informational content. This includes audio transcriptions, titles and meta data, as these combined give an accurate representation of the content of the videos.

Finally, we need to structure MOOC videos in a different way to encapsulate the multitude of sequence orders that can be followed after a video. We do this by setting all videos as nodes into a graph that are connected to each other with directed edges representing the order in which videos can be watched. With this graph of videos, we can easily provide alternative sequence orders per video, as all outgoing edges per video node give different sequence orders that can be followed. This allows an online learner to plan out their own personalized sequence order based on their interests and background.

1.1 Research Questions and Contributions

Based on the description above, we define several research questions. The contributions of this thesis are defined as the answers to these questions. The main research question is the following:

**Research Question 1.** How can we improve the way that online learners navigate through MOOC videos to help them find alternative videos suited to their needs?

We attempt to help online learners to find the right videos that suit their needs by adding alternative sequence orders of MOOC videos. For every video in a MOOC, we add alternative successor videos besides the original sequence order defined in the MOOC.

**Research Question 2.** What data structure should be used to store these new alternative sequence orders along with the original MOOC sequence orders, and how should this structure be made from a collection of MOOCs?

We start by placing the MOOCs separately into a graph. The videos from a MOOC can be represented as nodes, which are connected with directed edges representing the sequence order of the videos. When there is an alternative sequence order to follow for a video, a new edge is placed that is directed from the video node to the next video node in the alternative sequence order.

**Research Question 3.** Are the alternative sequence orders good alternatives to the sequence orders defined in MOOCs?

As it is hard to get an objective review of the alternative sequence orders that are created, we ask users to rate the alternative sequence orders. This is done by conducting a user study in which participants are asked to read the transcript of one MOOC video and rate the quality of a set of successor and similar videos. By doing this user study, we are able to evaluate the quality of the new sequence orders. The results of this evaluation can be used to compare it to the sequence order described in MOOCs and semi-randomly picked sequence orders. This allows us to get as close to an objective review of the alternative sequence orders as possible.

**Research Question 4.** Can these alternative sequence orders be created within a reasonable time frame when dealing with a large amount of data?

As there are a large amount of MOOCs, we need to ensure that our method is able to create a graph of videos within a reasonable time frame. A simple brute-force way of finding alternative sequence orders would be to simply compare all videos to each other, making for a quadratic number of comparisons. By carefully selecting the video comparisons, we can avoid a large amount of comparisons, reducing the run-time significantly.

We will go into detail about these contributions in the following chapters. We start by explaining the syntax used in this thesis in chapter 3. Chapter 4 explains the properties of the data which is used in this thesis. Before we go over the theoretical parts of this thesis, we conduct a pre-study in chapter 5 to examine our data set and establish some ground truths upon which we can build our theory. The design of the method that creates alternative sequence orders is discussed in chapter 6. In chapter 7 we will go over several ways of calculating a similarity rating between videos and how we can find similar videos with these ratings. Chapter 8 covers the solution to create extra edges between videos, and discuss the methodology on how we got up to that solution.
chapter 9.1 we conduct an experiment in order to find ideal similarity thresholds, and whether our methods actually returns accurate similar videos and successor videos. Finally, chapter 9.2 goes over the challenge of applying the methods in this thesis to practice, as it proved to be a major challenge to decrease the run-time to a feasible run-time.

2. Related Work

We categorize related works in three different groups: studies that contribute new ways of improving the content of a lecture, studies that analyze properties shared between MOOCs and studies that suggest methods of analyzing and comparing lectures.

2.1 Improving Lecture Content

From a top level perspective, this thesis aims to help online learners find their own preferred path of MOOC videos. We attempt this by adding alternative video sequence orders, but other studies have found different ways of improving the online learning experience by improving the way lectures provide content to online learners. These studies describe how video interaction data[8], key-term extraction[11, 20], summarizing lecture videos[7, 11] and labeling videos[15] improve the learning experience of online learners. These methods focus on taking an existing lecture and apply enhancements on the lecture based on its content and metadata. These methods allow an online learner to understand the content of the video in a better and quicker than just watching the entire video.

While these methods have great value to them, it does not provide any information regarding the sequence order between lectures. This is because their view is focused on improving a stand-alone lecture, while disregarding the context around the lecture. Because these studies focus on single lectures instead of a combination of these lectures, we do not regard these methods to be relevant in this thesis.

2.2 Analysing MOOC Sequence Order Relationships

Wang et al. did a relevant study on prerequisite knowledge relations as they attempted to find learning gaps by extracting domain key concepts and then identify prerequisite relations between extracted key concepts[25]. The key concepts were extracted from Wikipedia as its content is filled with other concepts related to it. They extract the top concepts from the Wikipedia article based on the similarity between it and the key concept. We apply the concept that Wang et al. uses to identify prerequisite relations, while we use a different methodology in order to apply this concept.

Yang et al. did a similar study by mapping Wikipedia categories to concepts from a university course graph[26]. When mapping multiple course graphs to Wikipedia categories, they could find relationships between course graphs where courses link to the same Wikipedia category. This allows us to combine course graphs together to increase the number of options a student can choose to learn. The way Yang et al. map course graphs to Wikipedia categories is relevant to this thesis, as we attempt to map lecture videos together as well.

Scheines et al. tested whether they could infer prerequisite structure based on variation in student performance on an assessment[22]. They designed an algorithm based on the Q-matrix, which maps the tasks that students perform with the required skill, that has a 64.5% true positive rate for discovering adjacencies. The goal of Scheines et al. is similar to the goals of this thesis, while their methodology is different. Scheines et al. base their algorithm on the Q-matrix while this thesis relies only on the data provided by the MOOCs.

Shen et al. looked at connecting videos based on the predecessors of similar videos[23]. The videos were compared using the audio transcripts and/or titles of the video. The similarity of the transcripts is obtained by comparing tf-idf[19, 21] vectors using the cosine similarity measure[21]. The similarity of the titles is compared by looking at the grammatical rules used in the titles, as well as by comparing the parsing trees of these grammatical rules and the similarity between words. They implemented a system which allowed a user to query a subject, and then created a schematic overview by linking any prerequisite videos to the queried subject. They combined videos that cover this subject from several organizations and created a map to provide a schematic overview that clarifies the learning situation for the given subject. Based on the work of Shen et al., we can conclude that it is beneficial for nearly all users to provide extra links between online videos. Their study shows that 97.4% of their users agree that such a map is useful in learning.

In contrast to this thesis, Shen et al. base their methods on the relation to predecessor videos rather than successor videos, and focus on creating a schematic overview of the queried videos. This thesis aims to help online learners follow an sequence order of videos that suits their needs and targets MOOC videos in general instead of a queried subset of MOOCs. We take inspiration from the structure and similarity methods that Shen et al. set up. This thesis will have a look at other similarity methods that are state-of-the-art.

2.3 MOOC Content Analysis

In order to find similar videos in MOOCs, we need to compare MOOC videos on the informational content that they provide, and we can achieve that by comparing the transcripts of the videos as shown by Shen et al.[23]. In this section we look at studies that compare methods for finding document similarity under different circumstances. We use these methods to compare video transcripts to each other to find similar video transcripts.

Where Shen et al. used tf-idf[19, 21] to compare transcripts, other studies showed different effective methods to
compare documents to each other. Word2vec[16, 17] and doc2vec[10] allow us to create frameworks that can effectively compare textual contents of lectures to each other. We can use these frameworks to create a vector-representation of the video’s content. The distance between these vectors of different lecture videos represent the dissimilarity of these videos. We use the cosine similarity measure[21] to compare vector representations to each other.

Camper et al. find that when comparing summaries of text to each other that doc2vec works best for finding document similarity[4]. They test this by computing document similarity with LSA, LDA, word2vec and doc2vec.

Charras et al. concluded a similar finding when comparing tf-idf and doc2vec for finding similarities in subtitles[5]. Where tf-idf managed to cover 66.3% of all unique utterances, doc2vec covered 87%.

Lau et al. made an extensive comparison[9] between doc2vec and word2vec, which are both word embedding techniques developed by Mikolov et al. The study of Lau et al. shows that doc2vec generally performs well and almost always outperforms word2vec when comparing documents for similarity.

Tao et al. compared tf-idf, word2vec and doc2vec for providing content feature for automated speech-scoring[24]. Doc2vec is also proven to be the superior method in this study, as it returns the the best content features when compared to human scores.

Maheshwari et al. had similar findings[14] regarding doc2vec. They evaluated their document similarity framework SimDoc against four different methods of measuring document similarity by trying to find similar research documents. While SimDoc managed to get an accuracy rating of 72.69%, doc2vec got an accuracy rating of 85%. They explain that doc2vec was able to get a better accuracy rating because it was able to compare whole documents to each other, while the other systems were limited to just comparing a part of the document.

Based on the comparisons in these studies, along with the findings of Shen et al., we will look at using tf-idf, word2vec and doc2vec as methods for comparing transcripts.

### 3.1 Schematic Syntax

MOOCs consist of videos categorized in collections of sections or lectures. In a MOOC, there is a described sequence order for the lectures. The lectures have a sequence orders for their sections, and the sections have a sequence order for their videos. When these video sequence orders are set after each other, we get the original sequence order for the MOOC’s videos. A schematic representation of this order can be seen in figure 3.

**Definition 1. **Original MOOC video sequence order - Each MOOC \( m \in M \) contains a collection of videos \( V_m \) which holds \( n_m \) videos. These videos are set in a sequence which is ordered from the first video \( v_{m,0} \) to the last video \( v_{m,n_m} \). This sequence of videos is described as the original MOOC video sequence order.

The only type of nodes that our graph will contain are video nodes. We will define this set of video nodes as \( V \). These nodes represent videos from a given MOOC, and these can be compared or connected to other video nodes.

**Definition 2. **Video node - a video node \( v \in V \) represents a video in a MOOC. We can compare and connect this node to other nodes from the set of nodes \( V \) with edges \( E \).

In order to create this structure, we start by representing the original sequence order of the videos as described in the set of MOOCs \( M \). We recreate this order by connecting the nodes in \( V \) in their original order defined in \( M \) with successor edges.

**Definition 3. **Successor edges - successor edges \( E_{su} \) are a set of weighted directed edges that are placed between nodes such that \( E_{su} \subseteq U \times W \cdot R_{su} \), where the nodes in \( W \) come next in a sequence order from nodes in \( U \). The variable \( R_{su} \) is a mapping of the unweighted edges in the set \( U \times W \) through a function \( f(v_1, v_2) \rightarrow r_{su} \) to produce a set of weights. This function takes two video nodes and maps these to a weight that is defined as the successor rating.

We now have a graph that contains the original sequence order of videos in MOOCs. The next step is to compare videos to each other to find how similar they are to each other. We do this with a similarity function that maps two video nodes to a similarity rating.

**Definition 4. **Similarity - similarity is represented as a function similarity\((v_1, v_2) \rightarrow r_{sim}\) that maps two video nodes to a numeric representation that represents how similar the two video nodes are. The function takes two video nodes \( v_1, v_2 \in V \) as input and returns a similarity rating \( r_{sim} \). The domain for this similarity rating ranges from 0, non-similar, to 1, identical.

We describe the similarity between videos in our graph by connecting video nodes with a similarity edge \( e_{sim} \). The weight of these edges represent the quality of the similarity relation.
Figure 2. An overview of the components that we use to describe a scenario schematically. The original MOOC video sequence order shown with successor edges, and the alternative sequence orders are shown with alternative successor edges.

Definition 5. Similarity edges - similarity edges $E_{sim}$ are placed between video nodes that are similar to each other such that $E_{sim} \subseteq U \times W \cdot R_{sim}$, where the nodes in $W$ are similar to the nodes in $U$. The variable $R_{sim}$ is a mapping of the unweighted edges in the set $U \times W$ through the similarity function to produce a set of weights. This results in $E_{sim}$ being a set of weighted undirected edges that connects similar video nodes with similarity edges that's weighted according to the similarity between the two nodes.

The final step to creating a structure is adding the alternative sequence orders to the graph. We represent these alternative sequence orders by connecting the sequences of videos with successor edges.

4. Data Properties

The MOOCs used in this thesis are gathered from Coursera[1] and Khan Academy[2]. Coursera and Khan Academy are among the leading providers of MOOCs, with Coursera alone providing more than 2000 MOOCs on a wide variety of topics. Khan Academy is rising up as one of the main providers of technically oriented MOOCs, as most of their MOOCs provide content on scientific subjects.

The data used in this thesis consists of 24 MOOCs that are carefully selected from Khan Academy and Coursera, with a total collection of 1626 videos. This amount of data is chosen to provide enough data to be able to experiment with finding alternative sequence orders, but small enough to keep the data set controlled, and allow for easy experimentation.

The MOOCs in this data set can be divided into three sets that each covered a field of study. The fields of study that are represented in this data set consists of computer science, chemistry and music theory. These fields are chosen to represent a variety of subjects that are clearly distinguishable from each other.

4.1 MOOC Structure

Both Coursera and Khan Academy have their data in a hierarchical structure as shown in figure 3. Per course, you have multiple lectures. Lectures have multiple sections, and sections have multiple forms of study material. Study material within the same section usually cover the same subject and are likely to be related to each other. Different sections within a lecture usually cover subjects related to each other, but not necessarily comparable to each other. Every level of this structure has an order assigned to them in the MOOC that the viewer should follow. The order in which the videos are placed are referred to as the original video sequence order of MOOCs, as described in definition 1. Figure 3 also shows the average ratio between these levels in the structure.

The study material mostly consists of videos, documents and tests. For the scope of this thesis, we will focus solely on the videos in MOOCs. This allows us to stick with one consistent form of data.

4.2 Video Data

A video contains three different types of data: video, audio and metadata. While video content is one of the forms of data, it is difficult to compare two sources of video content on its informational content. We can look at the actual video data, and compare them by analyzing them frame-by-frame. However, this results in a comparison of the graphical information, and not the informational content of the video.

The audio in the video holds a lot of informational content, as it portrays what the teacher is saying. Every video in the data set has a textual representation of the audio in the form of a transcription. Some organizations create these transcriptions manually, while others automatically generate transcriptions based on the audio. A problem that transcrip-
Figure 4. Durations of all videos used in the data set of this thesis. 96% of all videos are under 20 minutes long.

...tions can have is a lack of quality. The automatic generation of a transcription or a poorly done transcription can impact the accuracy in which we can compare it to other transcriptions. However, since this is out of the scope of this thesis, we assume that the quality of the transcriptions does not have a significant negative impact on our results.

Finally, every video contains some form of metadata of which several attributes are interesting. The attributes that are covered in this thesis are the video title and duration of the video, as they can be compared to the same attributes on other videos.

Based on the 1626 videos in our dataset, the videos have an average duration of 8.5 minutes. The distribution of video durations is shown in figure 4. The difference between video durations is mostly small enough, but some of the outliers will have a large difference in duration. However since 96% of all videos are under 20 minutes long, the amount of videos that have significantly different durations is low.

5. Pre-study

It is pointed out in previous chapters that the alternative sequence orders are based on the original video sequence orders in MOOCs, and the similarity between their videos. In this chapter we want to verify if we can base the alternative sequence orders on these original sequence orders. This is done by conducting an experiment to evaluate the quality of this original sequence order, so that we can safely base the alternative sequence orders on these original sequence orders.

A good sequence order of videos allows the online learner to be able to watch the videos without having a lack of presented information inbetween the videos. When a successor video cannot be watched because of a lack of information, this means that the first video did not provide enough information and thus makes the two videos set in a bad sequence order. However, since it is hard to objectively rate the sequence order of videos, we set up an experiment in which we ask participants to rate the quality of this video order. In this chapter, we cover how we set up this experiment and what results came out of this experiment.

5.1 Experiment Setup

We set up an experiment in order to evaluate the quality of the original sequence order in which videos are placed in MOOCs. This would give an indication of the quality of the successors that come in the order defined in MOOCs.

In this experiment, we asked 10 MSc Computer Science students to read the transcriptions of five videos and the successors of these five videos:

1. *Atoms and Elements*, in course *Intro Chemistry*
2. *Conditional Statements and Loops in Python*, in course *Machine Learning Foundations*
3. *Fundamentals Data Model*, in course *Machine Learning Regression*
4. *Major Scales*, in course *Guitar*
5. *Regression ML Block Diagram*, in course *Machine Learning Foundations*

These videos are chosen specifically because three out of five videos are related to Computer Science, while the other two videos are likely to be a less familiar subject for the participants. This is done so that there is no bias to subjects that the participants have knowledge of. The participants rate the quality of this sequence order of the video and its successor by giving a positive, neutral or negative rating.

5.2 Results

In total, we got 50 ratings from the participants in this experiment. These ratings are displayed in table 1. When analyzing these results, it became apparent that all the negative user ratings came from one specific case. When a video was shown with its successor being in the next section, we noticed a large increase in the number of negative user ratings. Table 1 shows a clear distinction between videos that share the same section as their successor, and videos that do not share the same section with their successors. When they do not share the same section, we can see that the number of negative ratings is significantly higher than other ratings, and need to be taken into account. When they do share the same section, the number of positive ratings is significantly higher than the neutral ratings, and make for a compelling argument that the successor is of good quality.

Overall, the ratings show that the order of these videos are of good quality. Without any negative ratings on the videos in the same section as their successor, this makes for a strong case to believe that these are ordered in a logical fashion.

When the final video in a section has a successor to a new section that covers a new subject, users might disagree on the order of these subjects, but this might make sense from the overview of the whole course. So because of the limited
<table>
<thead>
<tr>
<th>Overall</th>
<th>Positive</th>
<th>Neutral</th>
<th>Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall</td>
<td>0.76</td>
<td>0.16</td>
<td>0.08</td>
</tr>
<tr>
<td>Different section</td>
<td>0.40</td>
<td>0.20</td>
<td>0.40</td>
</tr>
<tr>
<td>Same section</td>
<td>0.85</td>
<td>0.15</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Table 1. Ratings of the quality of the original order in MOOCs. Percentages are shown for the given positive, neutral and negative ratings. The ratings on videos within the same section as its successor and in a different section from its successor are separated for each other to show the origin of the negative ratings.

view the participants get to see, they cannot correctly judge the correctness of going from one section to another in two videos.

6. Method

So far we have described the reasoning behind creating alternative sequence orders. This chapter explains how we design a method that is able to find and create alternative video sequence orders between MOOCs.

The goal of this method is to find alternative sequence orders based on a given collection of MOOCs defined as $M$. This collection is retrieved from organizations such as Coursera, as covered in chapter 4.

The method starts by filling all the videos from the set of MOOCs $M$ into the graph, for which we can fill in sequence orders. In order to find alternative sequence orders, we look for similar videos between the MOOCs and create new sequence orders between them. Similar videos are found by going over all videos in every MOOC, and compare these videos to the videos of the other MOOCs in the collection with the similarity function defined in definition 4. This way, all videos are compared to the videos of the other MOOCs. For every comparison of a pair of videos, we generate a similarity rating.

Once we have a similarity rating between all the compared videos, we filter out all video pair comparisons which are below a given similarity threshold. The remaining video pairs are then used to create an alternative sequence order.

The pseudocode in algorithm 1 shows the strategy taken to create alternative sequence orders and to construct a graph of videos and their sequence orders.

The two methods $\text{similarity}$ and $\text{createSequenceOrders}$ are filled in for now, and will be explained in detail in chapter 7 and 8 respectively. The $\text{similarity}$ method is used to compare $v_1$ and $v_2$ to find a similarity rating. The $\text{createSequenceOrders}$ method is used to create a new sequence order directed from $v_1$ to $v_2$, and is portrayed in the graph as a successor edge between the nodes of $v_1$ and $v_2$.

The method as it is described in algorithm 1 is not optimized, as the process of optimizing it will be described in detail in chapter 9.2.

Algorithm 1: Method of constructing a graph of videos and their sequence orders, and finding and creating alternative video sequence orders.

7. Similarity

In this chapter we go over the various ways of calculating the similarity between two videos that we use in our method for finding alternative sequence orders described in chapter 6.

To find these similar videos, we compare videos to each other by the three types of data that each video holds. We can use the meta data and transcriptions of the audio data to compare videos to each other (as covered in chapter 4). In order to know how similar two videos are, we calculate a similarity rating based on the comparison of their respective types of data. We use this similarity rating to be able to label videos similar to one another. We explain how we go from similarity ratings to similar videos in chapter 7.3.

7.1 Transcription similarities

There are several ways of comparing transcriptions to each other, which can give a good indication of the video similarity. We compare video transcriptions to each other by using document similarity frameworks that capture both syntactic and semantic information[21]. Based on related works (see chapter 2), we will look at the following methods: term frequency inverse document frequency (tf-idf)[21].
word2vec[16] and doc2vec[10]. We can use these methods to create a vector representation of a given transcription. We regard two transcriptions to be semantically similar if their corresponding vector representations lie close to each other according to some distance measure, and dissimilar if the vectors lie farther apart.

7.1.1 Tf-idf

Tf-idf is a method that quantifies the importance of a given word in a collection of words into a tf-idf value. These tf-idf values are calculated by looking at the frequency that a word occurs in its transcription (term frequency). This frequency is compared to the frequency of the word used in all the transcriptions in the same dataset (inverse document frequency). This results in a list of tf-idf values represented as a vector, which indicates the most frequently used words of the given document[19].

Using this, we can convert all words in a transcription to a tf-idf value, and represent the transcription as a vector of tf-idf values. We can use this vector of a transcription and compare this to the vector of another document. We compare the vectors with cosine similarity, which results in a value between 0 and 1, where 0 means that the vectors are not similar and 1 means that they are identical. Cosine similarity can be calculated on two vectors using the following equation:

\[
1 - \frac{u \cdot v}{|u||v|}
\]  

(1)

The benefit of using tf-idf is that we do not need any additional training data to use this. It is only reliant on the transcriptions that are given, and results in comparable values which we can use to indicate how similar two transcriptions are. However, by only looking at the frequency of the words being used, we lose the context in which the words are used in. This can give a skewed image and might lead to inaccurate similarity cases.

7.1.2 Word2Vec

Word2vec is a technique that embeds single word tokens from a corpus of documents into a vector space using a continuous bag-of-words or skip-gram model. From this vector space, one can retrieve a word vector that represents a given word in the context of the corpus[16]. Given a video transcription, we can feed all words into this vector space, and average the vectors out into a single vector, representing the entire transcription. We can compare this vector to another transcription vector using cosine similarity found in equation 1.

There is a distinct difference between word2vec and tf-idf. Word2vec preserves the context in which the word is placed in its corpus, while tf-idf removes this context and only looks at the importance of the word in its corpus. While both methods work well for comparing the syntactic meaning of a word in a corpus, word2vec also covers the contextual meaning of a word, which is of added value when the goal is to compare informational content of two transcriptions.

However, as a word vector carries over the context of the word in the corpus, we risk losing context of the transcription when we average all word vectors into a singular transcription vector.

7.1.3 Doc2Vec

Doc2vec is an unsupervised framework that learns continuous distributed vector representations for larger blocks of text, such as sentences, paragraphs or entire documents[10]. It uses either distributed memory (DM) or distributed bag-of-words (DBOW) to predict word vectors, which are based on the earlier mentioned CBOW and Skip-Gram methods respectively.

Doc2vec has a strong advantage over word2vec, as it allows for larger blocks of text compared to the single word token limitation of word2vec. Word2vec is required to combine its word vectors into a single document vector, while doc2vec takes this into account when learning new word vectors.

Out of these three different methods, we find that doc2vec is the most promising choice, as it provides a context-based approach while preserving the continuous representation of a whole transcription instead of single words. This finding is shared across several studies[4][5][10][24], and thus we use doc2vec to compare transcriptions to each other during evaluation.

7.2 Metadata Similarity

Every video is accompanied with metadata, which includes data such as the title and duration of the video. This data allows for comparison on the title and duration on other videos.

7.2.1 Title similarity

The title of a video does not fully represent the informational content that it holds, but can give an indication of what it is about. By looking at similar words being used in the video titles, we can say something about whether they are alike. We can compare this in a similar way as we compare transcription. The video titles used in the data set of this study have an average length of 4.8 words. This means that they do not qualify for the ‘larger blocks of text’ which doc2vec excels at, and will lack enough context for word2vec to work optimally. Tf-idf can still give a good indication of similarity with a little amount of words, and will most likely perform the best of these methods.

7.2.2 Difference in Video Duration

The comparison of the duration of two videos does not give an indication whether the two are similar to each other. However, when the two videos have very different durations, it can give an indication whether they are dissimilar. Therefore
we decide not to use this rating as a way of finding good similarity measures, but rather to remove the bad cases of similarity.

7.3 Combining Ratings

We now have three methods of calculating a form of similarity between videos. All three methods give insight on some part of the video being similar or not. By combining these three ratings, we can give a definite answer whether videos are similar. However, we do not know when a similarity between videos is valid by only basing it on these arbitrary similarity ratings. Only when every similarity rating is equal to 1 does it guarantee that the videos are similar, as defined in definition 4. This is because a similarity rating of 1 means that the two videos are identical on the compared attribute. When all compared attributes are identical, we can safely say that the two videos are guaranteed to be similar.

While we can make claims about the likelihood of two videos being similar based on the similarity ratings, similarity is a subjective matter that depends per person. This is why we need to evaluate similarity rating thresholds that match actual similarity cases. This threshold will represent a minimal similarity rating at which we find most cases to be actual similar videos. Since we have multiple ratings, we should find multiple thresholds to aim for.

The transcription and title similarity ratings will share a common type of threshold, as they both will have a optimal rating at which the number of positive ratings and negative ratings are balanced optimally. This balance aims to keep most similar ratings to point to true positive similarity cases, while removing as many as possible false positive results.

The threshold for the difference in video duration will be used to remove negative similarity cases. Since a large difference in video duration will tend to point to a negative similarity case, we use this to remove negative cases instead of picking out positive cases.

We find the values of these thresholds in our experiment covered in chapter 9.1.

8. Creating Alternative Sequence Orders

The method described in chapter 6 uses a dummy method createSequenceOrders which is explained in detail in this chapter. This method should create alternative sequence orders from two given videos based on a calculated similarity rating between the two videos.

If we have deemed two videos similar, we know that they cover the same subject. For the sake of explanation, we can combine these videos and portray these as a single node in our graph. The edges going to both videos are redirected to the combined node, causing it to have multiple incoming and outgoing edges. When we take a step back to look at the original video nodes, we see that they still share the edges with their similar counterpart. In the process of combining these similar videos, we created new sequence orders between the videos. The new successor edges are assigned a successor rating equal to the product between the similarity rating between the videos and the successor rating between the video and its successor.

Figure 5 portrays a scenario where there are two collections of videos, with node B and E portrayed as similar nodes in (a). Since they are similar, this means that they can be represented as a merged node B, E as they have the same content, which is portrayed in (b). If we take a look at the original positioning of the nodes in (c), the merge of B and E would create edges between \{A, E\}, \{D, B\}, \{B, F\} and \{E, C\}, as they start to share common predecessors and successor nodes. However, if we only add either new edges between the predecessor or the successor nodes, we would add the same number of paths between the two collections of videos and prevent the addition of unnecessary edges. Therefore, we choose to only add the new successor edges \{B, F\} and \{E, C\}.

Let us take a look at the scenario depicted in figure 6. The nodes \(a_2\) and \(b_2\), both titled Matrix Vector Multiplication, are connected with a similarity edge, as they have a similarity rating \(r_{sim,a_2,b_2}\) above the threshold. These two nodes are considered similar, meaning that we can connect the similar nodes to the others’ successor. This is done by connecting them with successor edges. The node \(b_3\) is the successor of \(b_2\) as was defined in the original sequence order of the Linear Algebra MOOC, and thus has a successor rating \(r_{su,b_2,b_3}\) of 1. We can create a new successor edge \(\{a_2, b_3, r_{su,a_2,b_3}\}\) with a successor rating of \(r_{sim,a_2,b_2} \cdot r_{su,b_2,b_3} = r_{su,a_2,b_3}\). The same process goes for the successor of \(a_2\), namely \(a_3\). We can create a new successor edge \(\{b_2, a_3, r_{su,b_2,a_3}\}\) with successor rating of \(r_{sim,a_2,b_2} \cdot r_{su,a_2,a_3} = r_{su,b_2,a_3}\).

This process can be described with the pseudocode in algorithm 2. This describes the function createSequenceOrders in which two new successor edges are created from two video nodes \(a_i\) and \(b_j\) toward the nodes \(a_{i+1}\) and \(b_{j+1}\), based on the similarity between nodes \(a_i\) and \(b_j\) and the sequence.

![Figure 5. Schematic process of adding successor edges to the successors of similar nodes. (a) shows the scenario of two collections with a similarity edge. In (b), the two similar nodes are merged into a single node. When returning to the original position in (c), we add two new successor edges portraying the merge of B and E.](image-url)
In this chapter we evaluate the method described in chapter 6 twofold: first we look at the quality of the newly created alternative sequence orders, and secondly we look at the similarity between the nodes $a_2$ and $b_2$. The two similar nodes are connected to each others successors.

order between node $a_i$ and $a_{i+1}$, and node $b_j$ and $b_{j+1}$. This creates two weighted successor edges and adds them to the set of weighted successor edges $E_{su}$.

1. Figure out the thresholds of transcription similarity ratings, title similarity ratings and video duration differences.

2. Get an indication of the quality of the alternative sequence orders.

Algorithm 2: Pseudocode of the createSequenceOrders method, which creates two new successor edges from the video nodes $a_i$ and $b_{j+1}$ and adds them to the set of successor edges.

This process of creating alternative sequence orders based on similarity is evaluated in chapter 9.1.4.

9. Evaluation
In this chapter we evaluate the method described in chapter 6 twofold: first we look at the quality of the newly created alternative sequence orders, and secondly we look at the performance and how we are able to reduce the run-time to a reasonable time without impacting the accuracy in a significant way.

9.1 Quality of alternative sequence orders
We have described how we can create a graph based on the original order of videos and the similarity between courses’ videos. However, we do not know when a similarity between videos is valid by only basing it on an arbitrary similarity rating. In this chapter we aim to find thresholds for all three similarity ratings described in chapter 7.

Next to this, we want to evaluate if the methods described in the previous chapters make for feasible results. By doing this, we can measure the validity of the newly added successor edges.

The goal of this experiment is twofold:

We can evaluate both goals in a single experiment by exposing participants to a series of video comparisons. We asked a group of 10 MSc Computer Science students to give a rating per comparison which we can analyze and evaluate the system with.

9.1.1 Experiment Setup
In this experiment, the participants are shown a similar setup as in chapter 5. One video is shown, which is referred to as the main video, which the participant compares to a set of videos that are rated similar to the main video, and a set of videos that are deemed to be good successors to the main video. The participants are asked to rate the quality of each similar and successor video with a positive, neutral or negative rating, which are referred to as user ratings. The set of similar videos consisted of the three most similar videos to the main video and one semi-randomly selected similar video. This semi-randomly selected video is randomly selected from a set of videos that have at least a similarity rating of 0.5 or higher compared to the main video. This prevents the comparison between two very clear non-similar videos, as we found the similarity rating of 0.5 to be the lowest value of having at least somewhat similar videos. The set of successor videos consists of the successor as defined in the original sequence order of the MOOC, and the videos with the two highest successor ratings, along with one semi-randomly selected video with at least a successor rating of 0.5 as described above. This experiment resulted in 314 user ratings.

9.1.2 Similarity Thresholds
To find similarity rating thresholds at which we can find good similarity results, we need to compare the positive to the negative user ratings and look at the similarity ratings. The similarity rating threshold should indicate a border in which most of the bad similarity ratings are cut off, while preserving the majority of the good similarity ratings.

In figure 7 we mapped the corresponding transcription similarity ratings using doc2vec from positive, neutral and negative user ratings next to each other for comparison. We can see a clear separation between the positive and negative
ratings in which we can find a good threshold. At a similarity rating of 0.60, we retain 79% of the positively rated similarity cases while removing 60% of the negatively rated cases.

Figure 8 shows the mapping of the similarity between video titles and their corresponding user ratings. As is visible in the figure, most of the neutral and negative ratings and a significant portion of the positive ratings are located under a title similarity of 0.3. This means that it is hard to set a threshold on title similarity that will strike a good balance between removing negative results while keeping positive results. We find an ideal ratio between positive and negative user ratings when we do not apply a title similarity threshold, as it creates too many false negative results compared to the number of true negatives when even setting a low threshold.

The difference in video durations shown in figure 9 does not give a good indication whether the videos are similar or not. The difference between the positive and negative ratings is small and therefore does not show a good separation between the two user ratings. However when videos have a word count difference higher than 1400 words, then that will most likely mean that the user rating is not positive. Only 3% of the ratings with more than 1400 words difference in word lengths will be positively rated, while 12% is rated negatively. We therefore set the threshold for video duration at a 1400 word difference.

We think that when comparing two documents where one is significantly longer than the other, it might be hard to objectively compare the two videos, and will cause the users to go for a neutral rating.

9.1.3 User Rating Review

We are interested in the quality of the alternative sequence orders. This is found by comparing the user ratings of these alternative sequence orders to the user ratings of the semi-random sequence orders and the original sequence order defined in the MOOC.

We can get an indication whether the quality is good when high similarity/successor ratings get better user ratings than the randomly selected videos. Table 2 shows the comparison of user ratings between the top three similar/successor ratings against the randomly picked video. Compared to the randomly selected similar videos, our similar videos are a factor of 6.2 times likelier to be have a positive user rating, and our successor videos a factor of 8.5 likelier. Based on these results, we find that our higher similarity/successor ratings are more likely to have positive user ratings. We can also compare the user ratings of the alternative sequence orders to the user ratings for the sequence orders defined in MOOCs. The pre-study in chapter 5 shows that 76% of the sequence orders defined in MOOCs have a positive user rating, compared to the 59% positive user ratings for the alternative sequence orders. With a difference of 17% it shows that the alternative sequence orders are some-
what comparable in quality, but not as good as the original sequence orders.

The correlation between similarity/successor rating and their corresponding positive user ratings shown on the right column in table 2 show similar results. These correlation values indicate that there is a noticeable positive correlation between all the transcription ratings and their corresponding user ratings. Based on these correlation values, we again find that higher similarity/successor ratings from our system result in a better user rating.

9.1.4 Creating successors based on similarity

In order to confirm whether the method of adding new successor videos is valid, we take another look at similar videos and their successors. According to this method, the successors of similar videos should have a successor edge coming from both similar videos. In terms of this experiment, that means that a similarity case with a positive user rating \( sim_{a_1, b_1} \) should also have a positive user rating \( succ_{a_1, b_2} \) between the videos and their successors, in other words \( p(succ_{a_1, b_2} | sim_{a_1, b_1}) \) should be as high as possible.

When testing this hypothesis \( p(succ_{a_1, b_2} | sim_{a_1, b_1}) \), it turns out that 74.2% of the successors do indeed have positive user ratings, 6.5% neutral user ratings and 19.3% negative user ratings. This shows that the majority of the alternative sequence orders confirm the assumption of adding new successor videos based on the similarity between videos.

9.2 Optimizing run-time performance

In theory, the graph of videos is created by comparing all transcriptions, titles and meta data to each other so that a similarity rating is calculated between all videos. When this is applied in practice however, it requires a impractically large amount of memory and time to calculate. The most significant amount of resources are spent to compare transcriptions to each other using doc2vec. This method requires every video’s transcription to be compared to all other video transcriptions, except for the videos in the same course as itself is in. This means that when there are \( m \) courses, there are \( \frac{m^2}{2} \) number of course-to-course comparisons, resulting in a runtime of \( O(n^2) \). This quadratic increase of run-time results in impractically long run-times when the number of videos are increased. This effect can be seen in figure 10.

This runtime of \( O(n^2) \) is problematic, as the large size of the Coursera dataset alone would already take about 402 days to create a network of. This is in no way a practical run-time, and needs to be addressed before this system can be implemented.

This problem can be solved in a couple of ways. We could only compare videos to each other in the same field of study, meaning that we have to select videos upfront, and group them per field of study. This would mean that the system needs supervising in order to run, and would eliminate all links between different fields of study. For instance, every field of study covered in the Khan Academy dataset has about 6500 videos, and takes roughly 12 hours to compute, which would result in feasible run-times.

Another way to reduce the run-time is to start comparing videos, and try to share results between videos that are significantly similar to each other. This would mean that only the videos which are not similar to any video that has not been compared yet should check its similarities. By sharing results among similar videos, we could see a drastic decrease in run-time. However, since it is hard to say whether two videos are similar enough to each other to share results, this could very well result in shoddy similarities.

Finally, instead of only allowing for video-to-video comparisons, we could compare two subsets of videos. For example, if we were to look at a course, and find the 5 most similar courses, we could compare the course videos only to the videos of the 5 most similar courses. This effect would cause the run-time to increase in a linear pattern, and would guarantee that the videos are only compared to the most relevant courses. Another advantage of this method is that it remains an unsupervised way of calculating similarities. We are not guiding the method towards a specific set of videos, as the 5 similar courses are calculated by using the document similarity method described in chapter 7. However, because we are taking an arbitrary number of most similar courses, there is a chance that there are some crucial video compar-

<table>
<thead>
<tr>
<th>Positive similarity</th>
<th>Top three</th>
<th>Random</th>
<th>Correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive successor</td>
<td>0.59</td>
<td>0.08</td>
<td>0.70</td>
</tr>
</tbody>
</table>

Table 2. Percentage of positive user ratings of the top three videos vs. randomly selected videos, for both similar and successor videos. Correlations are between the similar/successor rating and their corresponding positive user ratings.
isons that will be left out. It could be that the videos are only being compared in the same field of study, preventing the process from creating any links between different fields of study. However, we believe that this chance does not outweigh the advantages of this method. It reduces the quadratic run-time to a linearly increasing run-time (which can be seen in figure 11), and stays unsupervised at the same time.

Figure 11. This graph shows the advantage of using a method that reduces the run-time from $O(n^2)$ to a run-time of $O(n)$. When we increase the number of videos to a reasonably large dataset, this makes a significant difference.

Overall, the final method seems to be the best way to improve the system’s performance. It is able to reduce the run-time to a constant increase of runtime per video without reducing the accuracy by a significant amount.

10. Discussion

This study was meant to create a method that helps online learners find alternative video sequence orders across MOOCs that fits to their interests, needs and background. In this chapter, we look back at our results and discuss whether we reached the goals we set out to do.

10.1 Reflection on research questions

This section is to reflect on the research questions we asked ourselves in chapter 1.1:

1. How can we improve the way that online learners navigate through MOOC videos to help them find alternative videos suited to their needs?

We came up with a method to find alternative sequence orders across MOOCs. We did this by finding similar videos, and create new sequence orders to other videos based on their similar counterparts. This idea is based upon common sense, and was tried to be proven based on the user ratings from the user study. We see from the results that our alternative sequence orders give comparable but not as good user ratings to the sequence orders defined in MOOCs. The heuristic that we make is that we can create new video sequence orders towards the successor of the video’s similar counterparts. When testing this heuristic by looking at the user ratings, we find that 74.2% of the similar cases give positive user ratings, meaning that most of the cases give good results and confirm this heuristic.

2. What data structure should be used to store these new alternative sequence orders along with the original MOOC sequence orders, and how should this structure be made from a collection of MOOCs?

We find that a graph is able to store the MOOC videos and their corresponding sequence orders in a good way. As a graph allows for applying algorithms such as finding the shortest path between nodes, it is ideal to use for things such as creating learning paths for online learners. Such algorithms will allow for automatized ways of creating personalized learning paths for online learners, or other implementations that can benefit from this graph of videos.

3. Are the alternative sequence orders good alternatives to the sequence orders defined in MOOCs?

It is hard to give an objective answer whether these alternative sequence orders are good alternatives. We believe that it is an unrealistic goal to reach the same level of quality as the sequence order in MOOCs, as these are manually made by experts. From the user study we find that the alternative sequence orders are significantly better than semi-randomly picked sequence orders, and comparable but not as good as the sequence order defined in MOOCs. Because our results are in range of the quality of these original sequence orders makes us believe that there is a definite potential in this method of finding alternative sequence orders.

4. Can these alternative sequence orders be created within a reasonable time frame when dealing with a large amount of data?

We managed to reduce the run-time from $O(n^2)$ to $O(n)$ by limiting the number of comparisons. When we try to add a course to the network, we start by comparing the course to the existing courses in the network. We only connect the course’s videos to the five most similar courses, which reduces the run-time significantly without compromising too much on the accuracy of the system. We found this solution to be easy to implement and leave the quality of the alternative sequence orders unaltered in any significant way.

10.2 User study

The user study was performed to get an indication of the alternative video sequence order quality. This is done by comparing the alternative sequence orders to the original sequence orders, semi-random sequence orders. In theory this setup would provide an objective review of the alternative sequence orders, it turned out that the number of partici-
pants was lower than initially hoped for. This resulted in a review that is less reliable than hoped for, but still gives a good indication of the quality of these alternative sequence orders. The user study shows that the alternative sequence orders are significantly better rated than the semi-randomly picked sequence orders, and comparable but not as good as the original sequence orders defined in the MOOCs. We discuss the possibility of a larger scale user study in the future work section.

### 10.3 Cyclic Graph

While this is not part of the topics discussed in this thesis, the graph that is created has a tendency to be cyclic. After connecting all video nodes with the original sequence orders and alternative sequence orders, we end up with a graph that has a high probability of containing cyclic edges. While this does not hinder the online learner directly, as the graph is used to show online learners the possible learning paths they can follow and thus does not depend on an acyclic graph, it can affect any analyses performed on the graph. Since we do not discuss this on this thesis, we do not see this as a relevant topic. The focus is on providing the alternative sequence orders, and use the graph merely to store and provide these sequence orders to the online learner.

### 11. Conclusion

We described a method that is able to create alternative sequence orders between MOOC videos that allows online learners to navigate through MOOC videos in a way that is suited to their interests and needs.

We started this thesis with the heuristic that we can base these alternative sequence orders on the similarity between videos in different MOOCs. We found that this heuristic holds for 74.2% of the cases. An alternative sequence order is created by creating a new sequence order between one video to its similar counterparts’ successor video. We find that these similar videos are best found by comparing transcriptions using doc2vec and remove any similarity cases where the difference in video duration is too large. All these comparisons result in a similarity rating.

In order to use these alternative sequence orders in practice, we structure the videos and sequence orders in a graph. The videos are represented as nodes, and the sequence orders are represented as directed edges between the nodes. This preserves the original sequence orders defined in MOOCs while also allowing alternative sequence orders to be added between videos. This graph is the result of the method described in this thesis. This method is supported with an extended discussion of the design space for implementing it.

Our evaluation shows that our alternative sequence orders have a positive user rating of 58.7%, while semi-randomly picked successor edges have a positive user rating of 8.0% and the sequence orders defined in MOOCs have a positive user rating of 76%. This shows that the alternative sequence orders are significantly better than the semi-randomly picked sequence orders, and comparable but not as good as the sequence order defined in the MOOC.

Thanks to run-time optimizations, our method is able to construct a graph out of a collection of MOOCs in a $O(n)$ run-time. In order to achieve this, we opted to only compare MOOCs to the five other most similar MOOCs. When the number of MOOCs start to become higher, this will prevent the run-time to increase quadratically, as every MOOC would be compared to all other MOOCs.

### 12. Future Work

We are satisfied with the results that this study has brought so far. However, there are some points that could still be touched upon. In this chapter, we will go through some of the steps that can be taken to improve the results that have been achieved in this study to make better use of the methods that we have described.

#### 12.1 Data Variety

This study has been working on MOOCs only, and limits itself to the structure that is usually used in them. However, one can imagine that it is possible to apply the methods discussed in this study to different kinds of data. We can see a clear pattern in the requirements of the data, which makes us able to use this structure to define a global required data structure to which we can apply the methods of this study to create additional navigational features in data. As long as there is a pre-defined sequence order in the data that connects the nodes to at least one other node, and the nodes can be compared to each other in order to find a similarity rating, we can apply the same methods discussed in chapter 8 on any kind of dataset. These data structure requirements need to be defined and tested in order to use them on any dataset.

#### 12.2 User Study

The user study done in this study was performed on a smaller scale than we initially hoped on, as it is challenging to find the right people and platform to evaluate the quality of the alternative video sequence orders. In the future, we would like to see the user study performed on a larger scale, as we believe that a larger user study would results in a more accurate threshold. Since our audience consisted of a narrow field of study (mainly computer science students), the evaluation might result in a different outcome when the evaluation brings in more users from a variety of backgrounds. In an ideal case, the method described in this system would be applied on a live system which could continually collect user ratings and adjust the similarity thresholds where needed.

### 13. Acknowledgements

I would like to thank everyone that supported me during the period of my thesis. I was able to carry this through thanks to your help. Thanks to Peter and Ria for serving coffee aplenty and providing great times for taking a break. Thanks to Tim and Laura for offering their assistance in reviewing my thesis many times. Thanks to Jan and Bas for their help and humor during the time spent at university. Thanks to Christoph for giving invaluable advice on how to improve my thesis, even when it sometimes struck an unexpected nerve.
Finally, I’d like to give a special thanks to my mother, brother and Lilian, who all had my back through this process and reviewed my thesis several times, even when my thesis took longer than expected. I would not have been able to do this without your help.

References