

Lecture Butler - Teaching Reasonable Lectures from a Lecture Video Archive

Martin Malchow, Matthias Bauer, Christoph Meinel
Hasso Plattner Institute (HPI)
University of Potsdam
Prof.-Dr.-Helmert-Straße 2-3
Potsdam, Germany
{martin.malchow, matthias.bauer, christoph.meinel}@hpi.de

ABSTRACT

Lecture video archives offer a large variety of lecture recordings in different topics. Naturally, topics are described superficially, easily or detailed in different lectures. Users interested in certain topics have problems finding lectures describing a topic chronology from basic lectures to more detailed difficult lectures. The Lecture Butler is going to automatically offer e-learning students lectures for the topics of interest in chronological playlists. The approach is finding lecture information using title, description, OCR and ASR data. This data is indexed and searched by an in-memory database to fulfill the speed requirements for playlist creation. In the search results lectures are going to be ordered by lecture occurrence in the university semester time schedule or by given lecture level of difficulty. As a result students can automatically create playlists for their topic of interest in sequence of the lecture level. Hence, students are not overstrained by lectures when they start with basic lectures first. Basic lectures provide information to understand more complex lectures. The research shows that an automatic approach by adding the level of difficulty or university semester time table is going to show reasonable playlists to find topics of interest. This solves the main problem students encounter when they try to learn a topic step-by-step using recorded lectures. The approach will support and motivate students using e-learning opportunities.

Categories and Subject Descriptors

K.3.1 [Computer Uses in Education]: Distance learning; H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval—*Information filtering, Relevance feedback*

General Terms

Human Factors, Reliability, Documentation, Experimentation

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.
SIGUCCS'15, November 9–13, 2015, St. Petersburg, FL, USA.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-3610-9/15/11 ...\$15.00.
DOI: <http://dx.doi.org/10.1145/2815546.2815557>.

Keywords

Teleteaching; Tele-Lecturing; Distance Learning; E-Learning; OCR Search; Semantic Web; In-Memory Database; Lecture Video Archive

1. INTRODUCTION

Lecture archives have grown up for more than ten years in general. One of the first approaches was the tele-TASK lecture archive mentioned in 2002 [8]. During this time more than 5500 lectures in different areas were recorded. The idea of the Lecture Butler is to find the most appropriate lectures for a student's field of interest. Furthermore, the student should have the possibility to find lectures which are close to their field of interest so that they can understand connections and context of a topic more deeply. Finally, the level of difficulty should also be considered. When taking into account that the lectures are held from the first Bachelor semester up to the Master program it is obvious that the lectures have different levels of difficulty. The idea is to take this level of difficulty into account to show lectures in the order of their level of difficulty in order to let people understand the topic of interest from scratch and motivate them to go on learning. When starting with a lecture which is too hard to understand students would become demotivated quickly and would stop learning [4].

This paper is separated into 5 parts. In the following section, related work in context of this topic will be described to show other research approaches and to highlight ideas used for the Lecture Butler's realization. This section is followed by the Lecture Butler approach. In the Lecture Butler approach the topics "Level of Difficulty for Lectures", "Setting a Predecessor of Series and Courses", "Network of Connected Words", "Synonym Structure", "Search for Custom Lecture Collection", "Playback Lecture Results", and "Self Tests" are discussed to illustrate their feature and the implementation details. In the evaluation in Section 4 a student survey will be discussed. In this survey we asked 43 students who are familiar with the lecture archive to evaluate the Lecture Butler features. As a result it shows which features are most important for students and should be implemented in lecture archives. Furthermore, it also shows which Lecture Butler feature does not enjoy great student popularity. Finally, the paper ends with a conclusion of the whole Lecture Butler topic and exhibits impressions for future work which can improve the learning experience of students using lecture video archives.

2. RELATED WORK

Interesting related work for the Lecture Butler approach was a lecture archive search algorithm using OCR detection and in-memory database technologies [7]. This approach is able to find most reasonable lectures by analyzing the lecture slide content. This slide content of all lectures will be considered when a lecture search is performed. Furthermore, this approach can handle the full text lecture search in a reasonable time of around 300ms per request. Due to the fast response time and the full text analysis of the lectures this search approach is suitable for the Lecture Butler implementation.

Another interesting research result is the automated keyword extraction from OCR data [9]. This approach finds OCR keywords of the slides. The keyword ranking method for multi-modal information resources algorithm is based on a “term frequency-inverse document frequency” (tf-idf). This research result is used for the Lecture Butler approach to extract keywords for the creation of the network of connected words. This network will be used in the Lecture Butler to find keywords connected to the main field of interest. Due to the described precalculation of the network this approach uses the precalculated keywords for the OCR detection result. There are also other methods like the keyword extraction on topic-level for learning objects [3]. This keyword extraction is suitable for learning objects in a special area. Nevertheless, in the first evaluation of the Lecture Butler approach the first keyword extraction method [9] created more appropriate results. One reason is that it is focusing on OCR data and the topic variety of the lectures has a wide range, which makes it more difficult for a topic-level keyword extraction. Especially, due to the fact that the topics have to be selected for every lecture or series manually in the worst case.

Furthermore, the semantic of the extracted keywords has to be analyzed. This is necessary to create the dependencies of terms to other connected topics. As base the DBpedia network will be used to get structured data [2]. DBpedia offers the Wikipedia knowledge as structured data in RDF format. These semantic connections can be used for own semantic analyses. In the Lecture Butler use case the found keywords will be analyzed with the DBpedia data and links to other Wikipedia topics within the Wikipedia topic page of the keyword. Finally, topic related terms can be used for the Lecture Butler search.

Another approach to use semantic technologies in video lecture archives is the semantic extraction of meta data used on lecture slides [1]. This work will find links in lecture slides and make them directly accessible for the user with a simple click. This avoids having to type links from the lecture slide manually. Additionally, links are checked for errors and unreachable or unavailable pages. This is necessary due to the fact that not all links are recognized correctly by the OCR detection. Furthermore, RFC numbers, ISBN book numbers, and Digital Object Identifiers (DOIs) are automatically detected and processed so that users can access the full mentioned resources in a lecture with one click.

Finally, a lecture archive system should be mentioned here. tele-TASK is a product portfolio starting from recording the lecture as dual stream, using the post production tool to edit lectures to publish the videos in the tele-TASK lecture archive portal [5]. The dual stream contains one video with the lecturer and another video with the lecture

slide or desktop content of the lecturer’s computer. This video is recorded simultaneously and will be played back synchronously in the lecture archive with the dual stream video player.

3. APPROACH

The idea of the Lecture Butler approach is to find parts or full lectures for a student’s field of interest. Therefore different tasks have to be performed in advance. These tasks are to set up the level of difficulty for lectures, create a network of topic connected words, and create a structure of synonyms. To realize the final search of the Lecture Butler four different search options are available “Lectures only”, “Sections of lectures”, “Dive into the topic”, and “Dive deep into the topic”.

3.1 Level of Difficulty for Lectures

The level of difficulty of a course or lecture series has to be determined. Therefore, two options are discussed in this work. One approach is to set the level of difficulty by values between 1 (easy) and 10 (hard). Another approach with the same evaluation method is the idea to set the difficulty level by the semester the course is planned in the university schedule. For example, courses planned in the fifth Bachelor semester get a number five and courses planned in the second Master semester get a number eight. Due to different durations of Bachelor and Master studies at different universities these values have to be adjusted according to the common durations of the Bachelor and Master programs.

A major challenge is the requirement to manually edit every course and add the semester information. As a solution predecessor courses are introduced. For every course a predecessor can be set up and when accessing the information for a semester the chain of predecessors will be gone through and only the oldest lecture needs to have the actual semester information. This information can, in addition, be used in the course overview. It enables the users to see all previous courses easily.

3.2 Setting a Predecessor of Series and Courses

When searching for lectures in different courses we should put lectures in an order according to their occurrence and difficulty. To realize this ordering semester information or the level of difficulty is necessary for a lecture. Due to the properties of a lecture, lectures will be repeated every year as a complete course. This characteristic can be used to reduce the effort of setting up a semester for every lecture or every course. Only the first course needs information about the level of difficulty or the semester it will be taught. All following courses in the following years get the lecture information from a previous year or semester as a predecessor is set up already. When calculating the index for the Lecture Butler search the semester or level of difficulty can be saved for every lecture by gathering course information and finding the semester in the dependencies of the courses. Furthermore, this information can be used for browsing through the courses from different semesters in the regular lecture archive portal.

3.3 Network of Connected Words

For automatic creation of the network of connected words DBpedia is the selected information resource in this approach. To read DBpedia information SPARQL requests are used as customized SQL language for RDF Processing. For searching terms we use the DBpedia property “wikiPageWikiLink”. This property is shows resources which are mentioned in the Wikipedia page of the requested resource.

However, DBpedia does not offer the “wikiPageWikiLink” information at the SPARQL online endpoint to avoid performance issues for the online version of DBpedia. Therefore, the DBpedia files can be downloaded¹. In our case using DBpedia version 3.9 in English and German is sufficient. The two most important DBpedia packages for the creation of the network of connected words are named “page_links_en” and “page_links_de”. They contain all connections from one Wikipedia document to another. With this additional information the connected words can be recognized automatically. To run the DBpedia SPARQL Service locally Virtuoso² has to be downloaded and installed with the already mentioned DBpedia files. After setting up the server, a SPARQL request can be generated to create the network of connected words. An example of a small part of the network for “WWW” is shown in Figure 1.

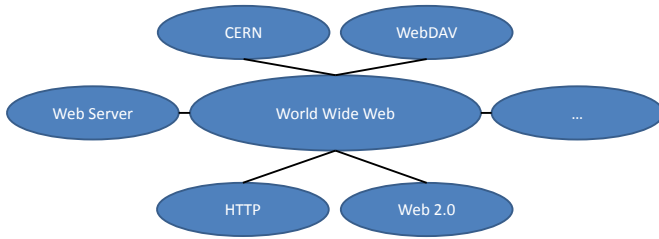


Figure 1: Network of Connected Words for the term World Wide Web

After finding all connected words in DBpedia, this information has to be stored for the Lecture Butler. Due to, the long execution time of SPARQL DBpedia requests the network of connected words are precalculated. The precalculated index will be stored in a database with the following two-table structure as network. One table stores all terms with a unique ID and a flag if it is a found keyword in a lecture or a keyword which was found with the help of DBpedia. The other table stores the connection between terms of the first table with the direction and a weight. Normally, the weight is just one as DBpedia does not store the number of connections to other Wikipedia topics. Nevertheless, the network is staff-editable and is just created automatically when a new lecture keyword [9] is indexed. Due to this behavior also the weight of a link can be edited by staff users to make some connections stronger than others. To persist staff-edited network settings there will be no re-indexing with DBpedia when a term is already in the term database and has the “found keyword in a lecture” flag. To handle issues with synonyms the approach uses an additional synonym structure described in the following Section 3.4.

¹<http://downloads.dbpedia.org/3.9/>

²<http://sourceforge.net/projects/virtuoso/>

3.4 Synonym Structure

In addition to the network of connected words described in Section 3.3 a synonym structure is necessary to build smaller networks to define what search terms have the same meaning. This is also necessary if different lectures use a different terminology for the same terms. For the realization a synonyms service like synonyms.net³ is used to automatically create a network of synonyms. This network is saved in a database like the network of connected words. There is one database table with all words and marked basic words which are used in the network of connected words called “Synonym”. Another database table is used for connecting the basic words with other words to mark synonyms called “Synonym_Relation”. With the help these two tables synonym information is stored. This information can be visualized like it is shown in Figure 2 for “World Wide Web”. The automated processed described with synonyms API is only used for the first creation of a synonym network. If the word already exists, no additional actions will be done automatically. Due to this implemented behavior users should be able to edit the network in the administration area without destroying this information in the database. Manual editing is reasonable because not all synonyms can be recognized correctly and completely by a computer in the process of automatic detection.

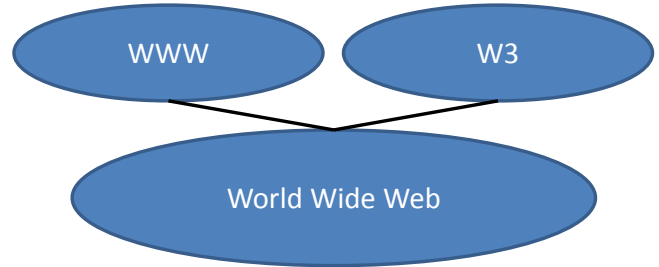


Figure 2: Synonym Network for World Wide Web

3.5 Search for Custom Lecture Collection

The most complex part of this work is the search of all data customized to users knowledge. The work flow to find suitable lectures is shown in Figure 3.

In the first step the user has to specify a search term, which is the topic of interest. Furthermore, the user can decide how deep this search is going to be performed. There are 4 search options available “Lectures only”, “Sections of lectures”, “Dive into the topic”, and “Dive deep into the topic”.

Using the “Lectures only” option a full text search is performed using the OCR slide search approach [7]. The results are limited to 10 when the in-memory database query is executed. Finally, the ten most appropriate lectures will be found and ordered by the semester as described in Section 3.1. The complete customized SQL query from [7] is shown in Listing 1. This query will select the ten most appropriate lectures and the semester if the semester information is available. The result of the ten most appropriate lectures has to be ordered by the semester information after getting the information from the database. To visualize the results

³<http://www.synonyms.net/>

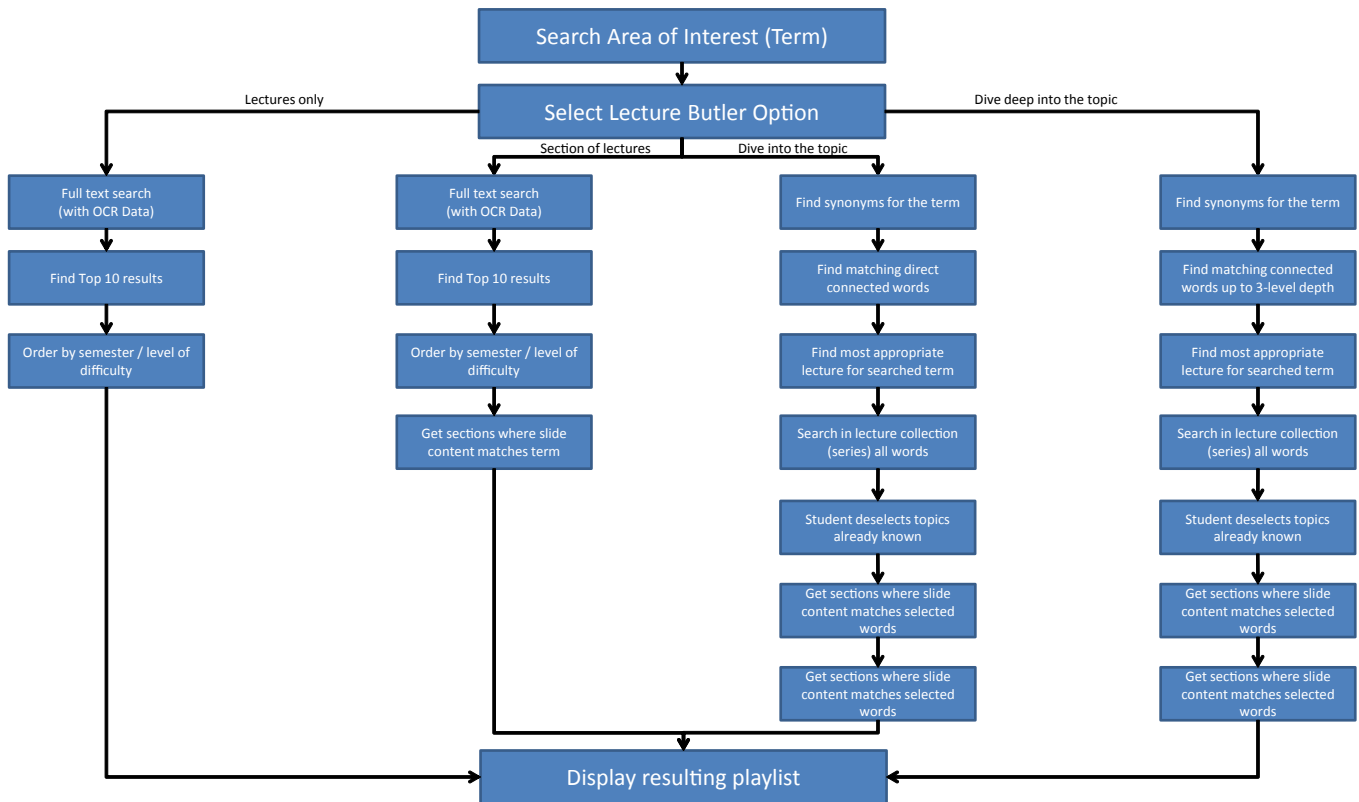


Figure 3: Workflow for Lecture Butler Search

the semester is shown in a progress bar to show how difficult it is to understand this lecture. Figure 4 shows an example of a search result using this approach. This approach also works if the semester is replaced with a level of difficulty for a lecture.

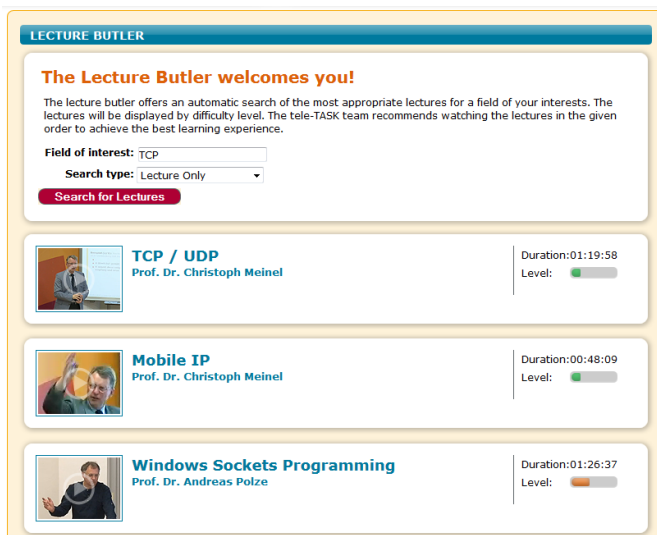


Figure 4: Lecture Butler search result for first option

Listing 1: HANA SQL statement to perform a lecture search selecting the newest lecture ordered by semester

```

SELECT SCORE() AS SCORE, T1."NAME",
T1."LECTURE_ID", T1."SEMESTER"

FROM "LECTURES" AS T1 INNER JOIN
(SELECT "NAME", MAX("LECTURE_ID")
AS "LECTURE_ID" FROM
"LECTURES"
GROUP BY ("NAME"))
) AS T2

ON T1."LECTURE_ID" = T2."LECTURE_ID"
WHERE CONTAINS
((T1."NAME", "OCR_TEXT"),
'searchterm', FUZZY(0.8)) AND "SEMESTER" != 0
ORDER BY SCORE DESC LIMIT 10
  
```

To perform a search for the option “Sections of lectures” the search goes more into the lectures by analyzing the OCR data of the slides for every section of a lecture. During the first step the search is performed like it is described for the option “Lectures only”. In the following step all OCR data for the sections of the ten most appropriate lectures is analyzed. As a result the ten most appropriate lectures are found with highlighting the parts of interest by sections. This improvement avoids watching sections which are not necessary for understanding the topic of interest. Furthermore, students are more motivated to watch the parts be-

cause it is not so time-consuming and the educational objective can be reached faster.

The third Lecture Butler option “Dive into the topic” enables students to learn a topic with contextually close topics. This means that students will get a broader overview of a topic to understand the connections with other relevant fields. To realize this Lecture Butler option the most appropriate lecture for a topic has to be found first. Additionally, information for the series of the lecture is loaded. In this context a series of a lecture is the pool of all lectures of a course. At this point, the network of connected words described in Section 3.3 will be used. This network of words can find topics which are connected to the field of interest. Now the topic of interest and all found terms in the first depth of one level in the network of connected words will be searched in all lectures of the series. Furthermore, the synonym structure described in Section 3.4 will also be searched for synonyms within the field of interest and connected terms. These additional synonym results are also considered for the search in the series. This leads to the next interaction step for the user. All connected words to the topic of interest which are also found in the pre-selected series will be shown on an overview page. At this point, the student can deselect terms he or she is already familiar with and which should not be shown in the final Lecture Butler result. Finally, the Lecture Butler shows all relevant lectures and the relevant sections with a description which field of interest topics and topics out of the network of connected words is discussed in this part. These results are ordered in the occurrence of lectures in the corresponding series. Figure 5 shows an example result page for this approach.

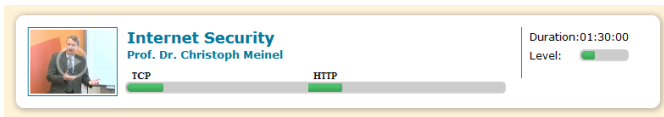


Figure 5: Lecture Butler “Dive into the topic” lecture result

Another option in this Lecture Butler paper is the use of the option “Dive deep into the topic”. The idea is similar to the one described in the option “Dive into the topic”. The main difference is the used depth in the network of connected words. Using this option neighbor topics in the depth of three will be discovered and not only the depth of one, like described in the previous option. This will lead to more results and enables the user to understand a topic more deeply with numerous of background information.

3.6 Playback Lecture Results

Typically the lecture results calculated like described in Section 3.5 can be played back in the standard player of the platform. In case of tele-TASK there are an HTML5 and a Flash Player. To handle jumps to special sections the URL time attribute can be used to start at the special sections part. Nevertheless, this basic idea has several drawbacks. When jumping to a special section and starting the video there is no possibility to stop the video after the section is finished. In case the search result offers sections two and four of a lecture as parts of interest the student will also watch section three and all following parts after section four.

Furthermore, when different lectures are in the result set of the search the student will watch a part and then has to go back to the result list and watch the next lecture result. This leads to a lot of manual effort for the students which could demotivate them using our Lecture Butler approach.

To avoid demotivated students the idea is to play the results one after the other automatically. For the realization HTML 5 video technologies will be used. The user can decide to play results completely. Now all found lectures and sections of lectures will be gathered in the right order. The video will start with the first lecture part in the given time frame. When this part is done there will be a split screen showing the lecture title, section name, and corresponding topics of the following lecture part. This will continue until all relevant lecture video parts are played back. As an extension of the player self tests can be added to enable students to check if the lecture content is memorized. This approach is described more detailed in the following Section 3.7.

3.7 Self Tests

Self tests give users the opportunity to check their knowledge after or while watching a lecture. An implementation idea and exchange format with a MOOC platform is used like described in [6].

In our environment we can import self tests from our MOOC platform and reuse them for the lecture archive or create own self tests. This self test will be shown after a lecture is finished in the normal player. When playing it in the HTML5 Lecture Butler mode after a lecture is showing information and the question is familiar with the topic of interest the question will be shown at the end before the following video will start like it is shown in Figure 6.

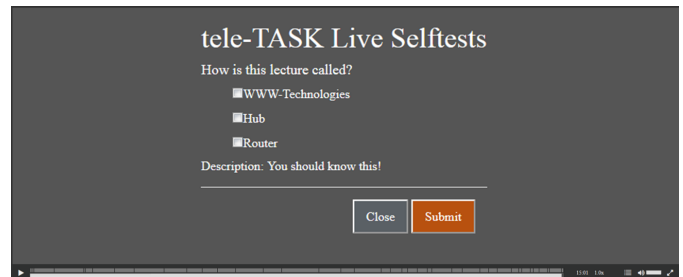


Figure 6: Lecture Self Tests in HTML5 Player

4. EVALUATION

For the approach evaluation 43 students who are familiar with the lecture archive answered several questions about the Lecture Butler. In the first step the students should search for a topic of interest. They then rated the value of the Lecture Butler search results. The results are visualized in Figure 7.

Like it is visible most students were satisfied or at least partly satisfied with in conclusion 93 percent. Only 7 percent of the students were dissatisfied with the results. When analyzing the search terms of users who are unsatisfied we noticed that they mostly searched for recordings they know which were not categorized as lectures. These recordings are categorized as colloquia. The Lecture Butler only searches for recordings categorized as lectures. Especially, to set the level of difficulty for the lectures according to the univer-

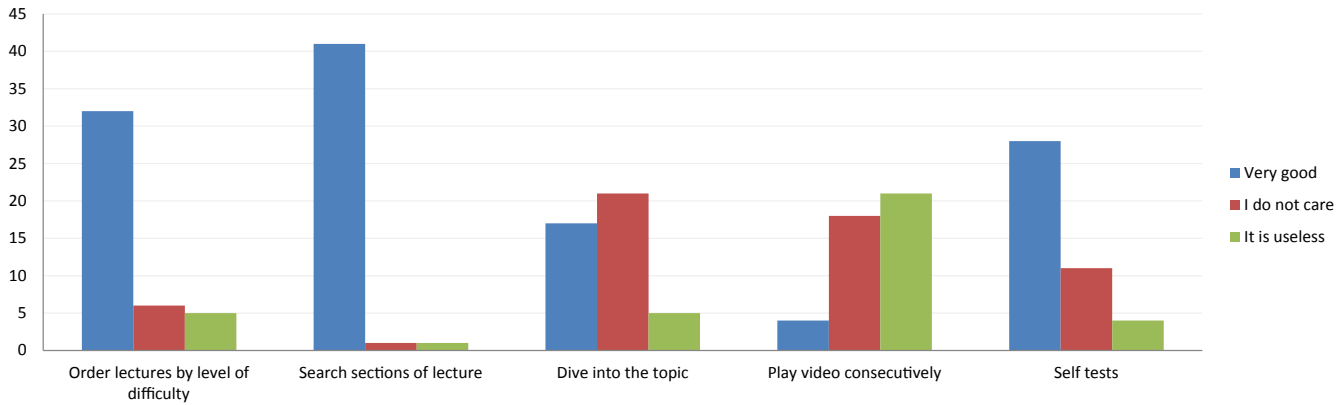


Figure 8: Student Evaluation of Lecture Butler Approaches

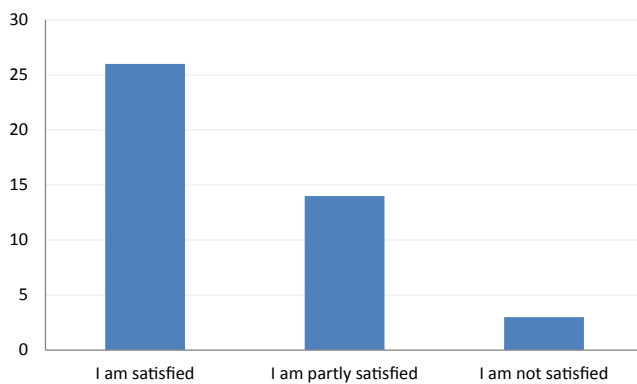


Figure 7: Student Search Result Satisfaction

sity schedule. Therefore, the results of the approach of the Lecture Butler are very good.

In the following the extensions of the Lecture Butler will be discussed. The evaluation results regarding the acceptance of the Lecture Butler’s additional features are visible in Figure 8. To avoid to overwhelm students with complex questions the evaluation uses a simple structure. It analysis if a student likes a feature (“Very Good”), they think the feature is not necessary but other students might need it (“I do not care”) or the feature is not necessary at all (“It is useless”).

The first feature the students were asked to evaluate was the ordering of the lectures by level of difficulty. 32 of the 43 students thought it is very useful to have an ordering by difficulty. Six students said it was not necessary and five students thought there was no reason to order lectures by level of difficulty. Some of the students said that they did not understand which rules this ordering is based on and thought it could be a subjective ordering by teachers. Therefore, the method for ordering lectures by level of difficulty should be described in the description of the Lecture Butler. When adding this information students understand how this ordering was done and trust the level of difficulty more if the ordering method is reasonable for students. In conclusion, ordering by level of difficulty seems to be an use-

ful approach, since most students attending the evaluation liked this Lecture Butler feature.

The following approach is “search sections of lectures”. Students also evaluated this feature. Students decided clearly that searching for sections of lectures is very reasonable functionality. 41 students liked this feature, one student thought it was not necessary and one student found this feature completely useless. As a clear result it is obviously helpful to use this Lecture Butler feature in lecture archives as students can find needed information fast.

“Dive into the topic” is the next considered feature of the evaluation. The majority (21) of students did not care about this feature. Only 17 students liked it and five students decided that it was useless. Due to a positive statement of this feature seems useful. Nevertheless, there should not be too much time invested in implementing this feature so more popular features shown in Figure 8 can be implemented with a higher priority. Furthermore, these results show that the option “Dive deep into the topic” is not necessary as students did not like the “Dive into the topic” very much. Furthermore, “Dive deep into the topic” will produce an enormous result with a lot videos and video parts. This might be very time consuming to watch.

The Lecture Butler feature to play videos consecutively in one player was not of interest to students. The students like to click on a video and play the clicked video back which is the result of the user evaluation. 21 of the students thought that this feature was useless, and 18 students did not care about a feature like this. Only four students thought it was a good idea to show all videos and video parts which mention the terms of the field of interest directly after one another.

The final student evaluation was performed in the self test context. 28 students decided that the self test approach was very useful while watching lectures. 11 students did not care about the self tests and 4 students thought that self tests were useless for their learning experience on lecture archive platforms. This result shows that it makes sense to implement this feature in the video player. Nevertheless, the player should have an option to disable self tests for students who want to focus on watching and not on answering questions.

The evaluation shows that most of the students were satisfied with the approaches and ideas of this paper. Except

when it comes to the consecutive video playback. Furthermore, not all features seem to be useful for all students using a lecture archive platform. To satisfy most of the users the features of the Lecture Butler should be optional. So students can enable or disable functionality.

5. RESULTS AND FUTURE WORK

This paper describes the approach to find reasonable lectures for a topic with the Lecture Butler. The Lecture Butler consists of several parts to realize a satisfying search experience for students. The first steps involve analyzing the lectures and must be done before the user executes the first search. These steps are to define the level of difficulty, setting the predecessor of series and courses, create the network of connected words, and create the synonym structure. When this process is finished the search can use the indexed data to perform the Lecture Butler search. The Lecture Butler search is separated into four options “Lectures only”, “Sections of lectures”, “Dive into the topic”, and “Dive deep into the topic”. Like the evaluation shows these options are necessary to fulfill the different requirements of students to give complete flexibility to search lectures with different strategies. Finally, the Lecture Butler offers different options for playing back the lectures out of the search result. The obvious playback possibility is directly using the player of the lecture archive. A more complex approach is to use a HTML5 player to show all relevant video parts in a sequence. Furthermore, a customized player can show self tests to remind students of important topics and help them think about the ideas of the lecture.

Even though the Lecture Butler is already very powerful future work has to be done. The evaluation result shows that especially students need a variety of options due to different learning behavior. The HTML5 player showing all video parts in a sequence should be optional and links to the normal lecture or lecture parts should be provided additionally. Furthermore, self tests should be disabled in the video player. This gives students the flexibility to decide freely about the learning experience. Finally, the basic search can be optimized by analyzing additional data like “Automatic Speech Recognition” (ASR) or by optimizing the search algorithm. Optimization can be done especially by indexing lecture information in the HANA. Currently, the bottleneck is gathering of lecture information from the MySQL database. Finally, a user analysis service should be added to the Lecture Butler to evaluate which options are mostly used and which topics are in the focus of students’ interest. With this additional information the Lecture Butler can be optimized for students’ needs and improve support and motivation.

6. REFERENCES

- [1] M. Bauer, M. Malchow, and C. Meinel. Enhance teleteaching videos with semantic technologies. In V. L. Uskov, R. J. Howlett, and L. C. Jain, editors, *Smart Education and Smart e-Learning*, volume 41 of *Smart Innovation, Systems and Technologies*, pages 105–115. Springer International Publishing, 2015.
- [2] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. Dbpedia - a crystallization point for the web of data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3), 2009.
- [3] K. H. Coursey, R. Mihalcea, and W. E. Moen. Automatic keyword extraction for learning object repositories. *Proceedings of the American Society for Information Science and Technology*, 45(1):1–10, 2008.
- [4] J. Falout, J. Elwood, and M. Hood. Demotivation: Affective states and learning outcomes. *System*, 37(3):403 – 417, 2009.
- [5] F. Grünewald, H. Yang, E. Mazandarani, M. Bauer, and C. Meinel. Next generation tele-teaching: Latest recording technology, user engagement and automatic metadata retrieval. In A. Holzinger, M. Ziefle, M. Hitz, and M. Debevc, editors, *Human Factors in Computing and Informatics*, volume 7946 of *Lecture Notes in Computer Science*, pages 391–408. Springer Berlin Heidelberg, 2013.
- [6] M. Malchow, M. Bauer, and C. Meinel. The future of teleteaching in mooc times. In *Computational Science and Engineering (CSE), 2014 IEEE 17th International Conference on*, pages 438–443, Dec 2014.
- [7] M. Malchow, M. Bauer, and C. Meinel. Enhance lecture archive search with ocr slide detection and in-memory database technology. *Computational Science and Engineering (CSE), 2015 IEEE 18th International Conference on*, Oct 2015. to appear.
- [8] V. Schillings and C. Meinel. tele-task: teleteaching anywhere solution kit. In *Proceedings of the 30th annual ACM SIGUCCS conference on User services*, pages 130–133. ACM, 2002.
- [9] H. Yang and C. Meinel. Content based lecture video retrieval using speech and video text information. *Learning Technologies, IEEE Transactions on*, 7(2):142–154, April 2014.