Experiences from a Pilot Project to Efficiently add Subtitles to an Open Source Lecture Recording Environment

Herwig Rehatschek, Marie Moriz

Executive department for teaching with media, Medical University Graz, Auenbruggerplatz 2, 8036 Graz, Austria

Herwig.Rehatschek@medunigraz.at, Marie.Moriz@outlook.com

Abstract. With the availability of affordable high-quality recording hardware and video management platforms lecture recording becomes a popular service for students at a steadily increasing number of universities. Since each university has its own infrastructure and general conditions, the introduction is still an individual process requiring a lot of technical know-how and a clear definition of the workflow process. At our university, we currently have about 740 recorded lectures, providing students access via our Learning Management System (LMS) Moodle and via an open source video portal. With a main focus on barrier-free access to learning material in general and hearing-impaired students in specific, we wanted to provide subtitles for all recordings. In addition, subtitles are also very helpful for students who do not have German (our main university language) as their mother tongue. Subtitles should be at least in German, preferably also in other languages in order to support foreign students (such as Erasmus) at our university as well. In this paper we will share our experiences how to efficiently create subtitles in a semi-automatic way. Furthermore, we will share the lessons learned with the introduction of the Open Cast platform and which technical workflow we particularly defined. This workflow is optimized for a moderate growth of recorded lectures – hence feasible for small and medium sized universities – and ensures a maximum of quality. It can be easily adapted to other universities.

Keywords: automated lecture recording, closed captions generation, recorded lectures.

1 Introduction

1.1 Background

With the availability of affordable high-quality recording hardware and video management platforms lecture recording becomes a popular service for students at a steadily increasing number of universities. Since each university has its own infrastructure and specific requirements, the introduction is still an individual process requiring a lot of

technical know-how and a clear definition of the workflow process. At our university, the introduction took three years, including one pilot year [1], [2], [3]. In August 2019, we finally could establish the final video management system and playout platform with the Open Source product Open Cast Matterhorn [14]. We currently have about 740 recorded lectures in our system, providing students access via our Learning Management System (LMS) Moodle and via a video portal [4]. With a main focus on barrier-free access on learning content in general and on hearing-impaired students in specific, we wanted to provide subtitles for all lecture recordings. In addition, subtitles are also very helpful for students who do not have German (our main university language) as their mother tongue. Subtitles should be at least in German, preferably also in other languages in order to support foreign students (such as Erasmus) at our university as well.

1.2 Purpose and Goals

However, creating subtitles manually is a human resource intensive process consuming a lot of work time. Hence, we performed research in order to save human resources on one hand and maximize the quality of the closed captions on the other hand.

In this paper, we want to share our experiences how to efficiently create subtitles in a semi-automatic way. Furthermore, we want to share the lessons learned with the introduction of the Open Cast platform and which technical workflow we particularly defined. This workflow is optimized for a moderate growth of recorded lectures – hence feasible for small and medium sized universities – and ensures a maximum of quality. It can be easily adapted to other universities.

For hearing-impaired students, subtitles are essential in order to efficiently study with recorded lessons. Besides that, not all students have German (main study language at our university) as their mother tongue, subtitles significantly help to improve the level of understanding. This is also true for environments with background noises (e.g. when watching the lectures on a mobile device on the bus) or you do not want to disturb other people and earphones are not available.

We had three main research issues:

1. efficiently creating subtitles for about 170 recorded videos, which corresponds to about 200 hours of video, and a moderate growth
2. finding a standardized format for the subtitles, which is human-readable and easy to be edited and
3. creating an adapted technical workflow, which enables the seamless integration of the subtitles into our existing open source video management and playout system.

2 Technical and Organizational Solution

In connection with the first research issue, we identified a number of commercial services, which offer a professional speech-to-text-conversion. In comparison to a simple creation of a text script, you also have to consider time codes in connection with subtitles. This is because each piece of text (referred to as “cue”) has to be synchronically

placed at a specific time in the video. This leads us to research issue (2) finding a proper standardized format for representing the text and the corresponding time codes.

After a short market research, we found the format WebVTT [5], which is a W3C recommendation. It fulfils our requirements: it is a human-readable ASCII based format, hence it can be easily read and edited with a text editor by a human and it can be imported into our OpenCast platform to be displayed by the Paella Player [6].

We solved research issue (3) by expansion of our existing technical workflow by adding automatic audio improvement [7], splitting the audio during import into OpenCast, and sending the audio file to a transcription service. Meanwhile, the two uploaded video streams (teacher video with audio, transparencies / PC output) are published on our video portal and via LTI on our LMS Moodle. After return of the audio transcript a manual improvement step is added, then the finalized subtitles are added to the already published video and are immediately available for the students.

Before we focus on the set of tools used in order to efficiently produce and process subtitles, we will first explain our technical workflow for the production of recordings. This will provide a good overview how we apply all necessary steps in practical life, how they are integrated in our infrastructure and into the open source Software OpenCast before we go into detail on the specific steps necessary for the subtitles generation. This workflow can be easily applied to other universities.

2.1 Technical Workflow

In **Fig. 1** the entire lecture recording process is visualized. The first step is pre-processing which contains the scheduling of the recordings. Since we have a central planning of the curriculum at our university, we ask teachers to provide recording wishes in advance. The lessons are then scheduled in the appropriate rooms equipped with the recording hardware. Additionally, we also provide requests on demand – in this case we try to re-schedule the lesson to one of the recording rooms – and recordings during live lessons. In the later we get a message of a new recording and contact the teacher right after the lesson for further steps.

Preprocessing and recording.

We decided to equip our lecture rooms with Epiphan [13] hardware, which met our technical requirements. All in all, we equipped five big lecture rooms, the aula, three seminar rooms and our clinical skills simulation center with recording hardware and with a camera. The recording interface can be easily controlled via a touch panel directly placed in the lecture rooms. The camera films the white board and the teacher, next to this the PC/Beamer output can be recorded. Both streams are recorded in full HD resolution. Furthermore, we provide the teachers with four recording pre-settings. The first setting records PC/Beamer and the teacher's lectern, the second setting records PC/Beamer and the teacher's lectern and the whiteboard, the third setting records PC/Beamer and the whiteboard and the fourth setting records whiteboard and teacher's lectern but not the PC/Beamer. These easy to understand recording scenarios together

with the record, stop and pause button – see **Fig. 2** - is the entire interface for the teachers in order to fully automatically record their lessons.

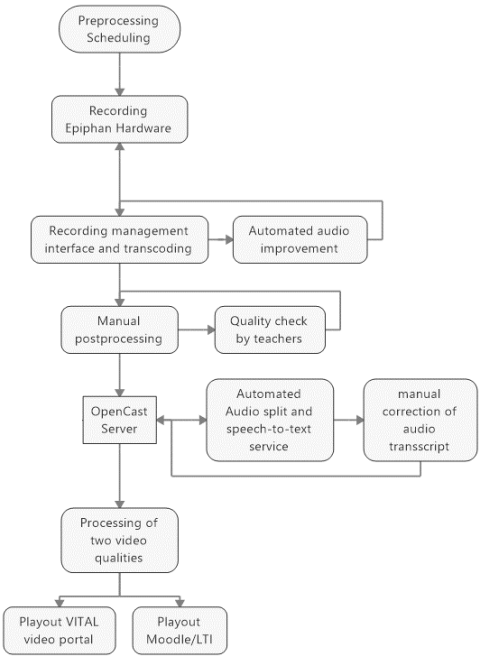

Fig. 1. Technical recording workflow including subtitle generation

Fig. 2. Teacher touch panel interface for recording management

Recording management interface and automated audio improvement.

After recording is done the data is automatically transmitted as a MPEG-2 transport stream containing both video streams synchronized to a storage. Here we programmed an interface which allows administrators to manage and control the recordings, teachers are decoupled from the process after recording and will be integrated only one more time later in the workflow for the quality control. The interface – see **Fig. 3**- supports the following functionality: notification per E-mail when new recordings are available,

download of the streams separately as a ZIP file or side-by-side, automatic metadata of the lesson (room, name of teacher, name of module and lesson, time of recording), remote control of the recording devices and archiving functionality. Here also the first step in connection with the efficient subtitle generation is built in: the automated audio improvement. A good audio quality is not only essential for students but also for the speech-to-text service, hence we transmit the audio automatically to an external service and deliver already the improved audio via the recording management interface. The service we have chosen due to good experiences with achieved quality and with reasonable pricing is Auphonic [7]. The service is a little bit faster than real time, meaning that processing takes about the same time of the video length. Hence, the video is available with a small delay for further processing and publishing, which is on our site no problem and shouldn't be for most other universities either. Following the defined workflow, we also use this interface for downloading the recorded material for the post-processing step.

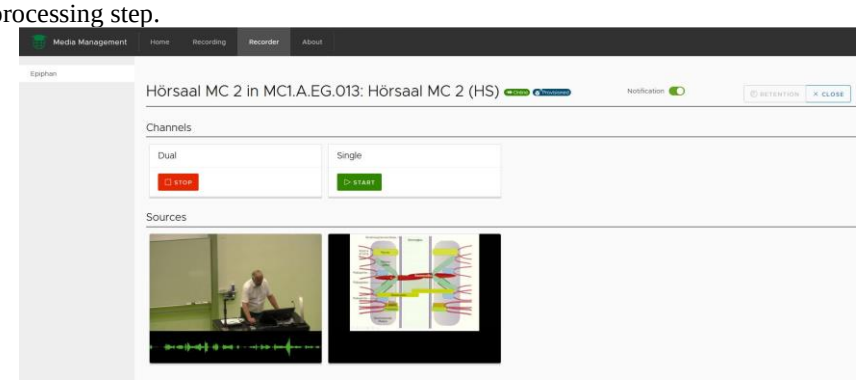

Fig. 3. Recording management interface for administrators with preview and remote control

Post processing and quality check.

In the post-production step we edit the recorded material. Due to the moderate number of recordings this step is currently done manually. We use Adobe Premiere in order to add a short introduction containing name of the module, title of the lesson and name of the teacher. Furthermore, we cut out sequences of bad quality, sometimes we add text bubbles when students ask questions without using microphones. Then we import the videos into our OpenCast server which renders two video qualities (Full HD and an SD). The video is published on the video portal where first only the teachers have access. The teacher performs a quality control of the ready-to-publish material. All wishes of the teacher will be taken into consideration – which may involve a re-editing and re-publishing - until they give their ok for publishing to the students.

Automated audio split, speech-to-text and manual subtitle correction.

When uploading the by the teacher approved material a special OpenCast workflow is utilized, which does not only generate the videos in two resolutions and publishes it on the portal and for our LMS Moodle, but also splits the audio and sends it to our speech-to-text service Amazon Transcribe [8]. Afterwards a manual correction step is

inserted, the by a human approved subtitle file is then manually added to the published video. Since this takes a while - see next chapter for more details – the video is published automatically and immediately after processing the two resolutions, subtitles are made available later on.

Playout to video portal and LMS Moodle.

The challenge for the playout software was to provide a user interface capable of playing two HD streams synchronically and to give students the flexibility for scaling the size of the two streams. In case the teacher shows something interesting on the whiteboard the video with the teacher can be zoomed and in case the information is only on the slides the video of the teacher can be switched off or made smaller. Furthermore, the player works in all standard web browsers without having to install any plug-ins and independently of the underlying operating system. Having all these requirements in mind we chose the Paella Player [6] which is now also part of the OpenCast open source software. We decided to offer two main possibilities to access the recorded lectures: 1) access via our LMS Moodle in order to access specific videos of specific modules, 2) access via a video portal where you have all recorded lectures in place ordered by courses and with a search function.

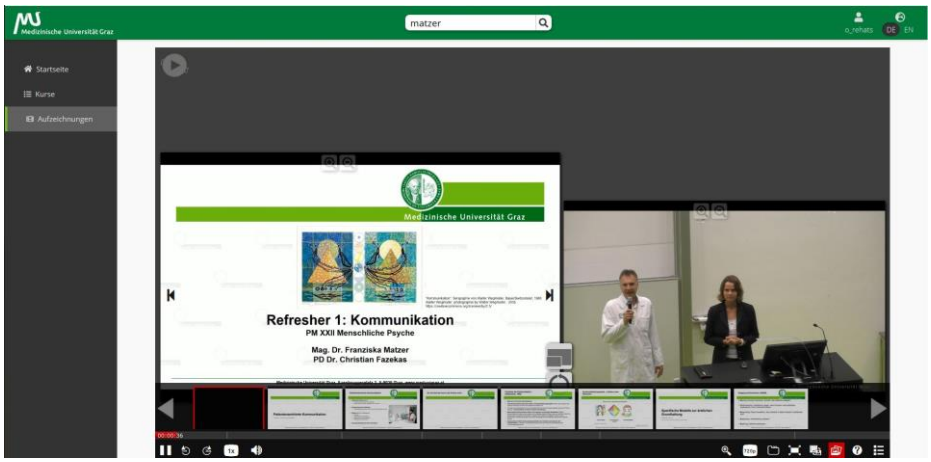

Fig. 4. VITAL video portal

In **Fig. 4** the portal solution VITAL and the Paella Player is depicted. Next to the above described functions the player offers also a variety of speed (in case the teacher speaks to fast or too slow), a simple slide segmentation for navigation, the selection of video quality, different layouts, and – most important with the subject of the paper – also the provision of subtitles in multiple languages. The portal is also open source software based on the lecture interface from University of Halle [16] and allows access to all our recorded lectures (447 by April 2020). In connection with access rights we defined two groups, students and affiliates, which are assigned to the uploaded videos and grants access to the corresponding groups. Besides this of course access can be

granted also to individuals or for the general public in case of open educational resources. The portal offers a hierarchical ordering by courses and provides a powerful search.

Integration into our learning management system Moodle is achieved via an embedded version of this player utilizing the standardized LTI (learning tools interoperability) interface [17].

2.2 Usage of tools for speech-to-text and subtitle editing

Based on the requirement to automatically produce a human readable text format which can be easily corrected by a human and integrated into our OpenCast platform we performed a small market research on speech-to-text services and tools which allow the manual correction of the produced WebVTT files.

Speech-to-text services.

For the transcription service we had the requirement that it must produce a WebVTT file as an output. In this connection there exists a vast number of available free and liable to pay solutions. There is a vast number of speech-to-text services available, free and commercial ones. As a free service you can use e.g. YouTube which will automatically perform a speech-to-text conversion and also provides a WebVTT compatible editor for the subtitles, see **Fig. 5**. Since we have our own video portal and storage we did not go for YouTube, which requires the upload and storage of the files on their server. Since our files have up to 10 GB, and also due to copyright issues this was no feasible solution for us. Also, the speech-to-text quality was rather poor for German language.

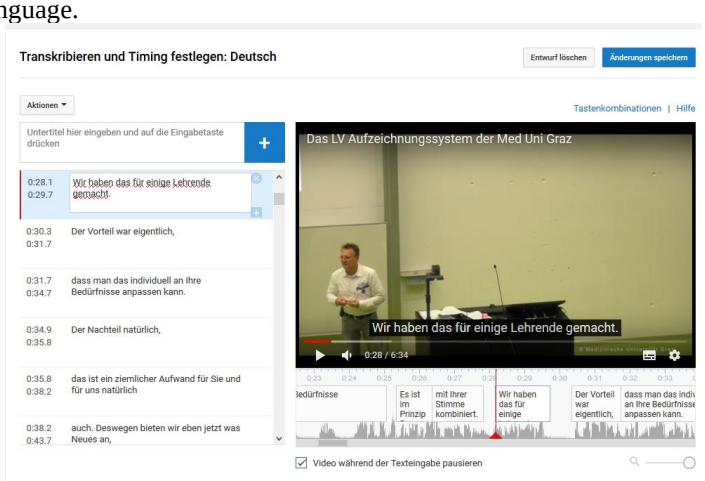

Fig. 5. YouTube closed caption interface

Next we tried out the commercial service AWS amazon transcribe [8]. Prices of the Amazon Transcription service for speech-to-text are reasonable even for low-budget universities having a moderate increase of video material. The service costs $ 0.00125 per second ($4.50 per hour), charged with an accuracy per second. In the first year, each

month 60 minutes are free. For the transcription of the first 170 recorded lectures we paid less than $140. The service already delivers the required WebVTT format including the time codes, which can be directly imported into Open Cast or to other platforms / players, supporting subtitles such as YouTube. We finally decided to go for this commercial service, because it also offers a module specialized on medical language (amazon transcribe medical) and provides more than 30 languages. Since we are a medical university, this service fulfilled our requirements. If you do not require the medical vocabulary, the service is even cheaper ($0.0004 per second / $1.44 per hour).

WebVTT editing tools.

What we have seen from the automatically generated transcripts so far: it is absolutely necessary to let a native speaking human proofread and correct the transcripts, at least when you plan to display them. In case you will only use them for searching, you might be satisfied with the delivered quality. Therefore, you will need a tool to process it efficiently.

For this purpose we again performed a small market research and found three solutions: the online tool VTT Creator [9], the subtitle editor from YouTube [10] and the very simple combination of a text editor (in our case Windows Edit) and our Video Portal VITAL.

VTT Creator offers a free, web based solution, which does not require any local installations. It has an integrated speech-to-text service, and offers a quite comfortable editor for correcting / adding subtitles, which is depicted in **Fig. 6**. The final file can be exported in WebVTT format for further usage.

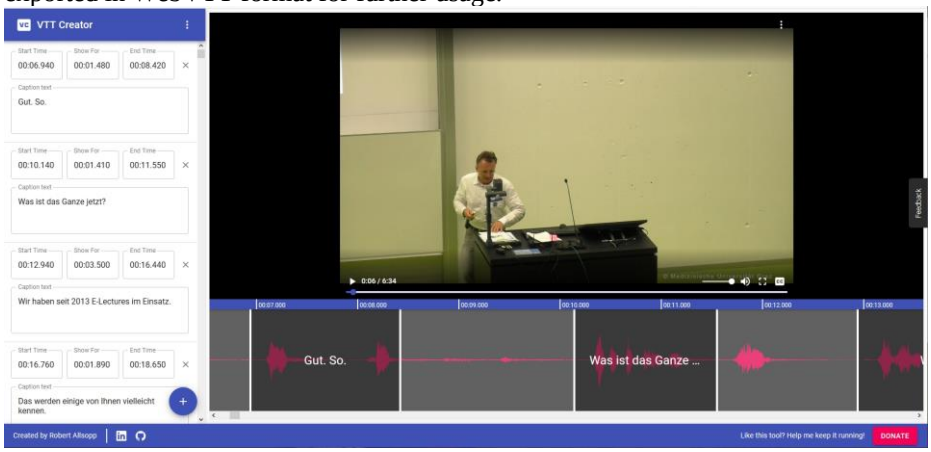

Fig. 6. VTT Creator online subtitle editor

For us the tool had several drawbacks: first the speech-to-text service is only available for English language (our main study language is German), second you have to upload the video files on the platform in order to take advantage of the service. Since our files have a size of up to 10 GB, the upload took a very long time, hence this solution was not feasible for us. We also measured the time it takes, to manually correct 10

minutes of a video transcript. All in all this took 77 minutes, also caused by the quite time consuming navigation within the user interface. In comparison to the finally by us chosen solution (see below) this takes more than 3 times longer.

YouTube also offers a subtitle service, which is depicted in **Fig. 5**. YouTube offers a speech-to-text service for a vast number of languages including German, however, the quality – especially in connection with medical terms – was poor. Again, an upload of the files is required, which is due to the size of the videos and in case of YouTube also copyright issues not feasible for us.

Next we tried to find free available offline WebVTT editors. In this connection we found Aegisub [11] and DivXLand [12]. Aegisub is a free open source tool, however, development was deceased in 2014, hence it does not support WebVTT format, which did not exist at that time. DivXLand offers only a Windows 8 version, but no Windows 10 version. The Windows 8 version does not smoothly work with Windows 10, so we skipped both options quickly.

Finally, the simplest solution turned out to be the best one. We used a simple text editor – in our case Windows edit, but any other ASCII editor would do it as well – in combination with our video portal VITAL (see **Fig. 4**). The person correcting the by AWS Amazon transcribe automatically created files opened them in a text editor and used the time codes in the file in order to navigate with the VITAL player to the positions in the video, which were not correctly transcribed and corrected the text directly in the WebVTT file. In order to better understand the spoken words a headset is highly recommended. With this process you have to take care only not to destroy the time code sections of the WebVTT file, hence only editing the text itself. However, since the format is very simple, this can be easily explained to the correcting person.

3 Experiences and Results

In general, we can say, the transcript quality significantly varied between a live recording (with students in the lecture room) and a recording made in an empty lecture room. We assume the main reason for this is the much higher background noise in a full lecture room made by the students. Hence, the manual correction effort in transcripts created from live recordings was higher.

The very best automatically generated transcript quality you receive by having a person recorded in an empty room with no background noise reading directly (and monotonously) from a script. Since this is a clear contradiction to doing a good presentation (it can be compared by a speaker who only reads from a paper and does not speak freely), we can state that technically it will be never possible to achieve 100% (or even close to 100%) accuracy with the automated speech-to-text process. Manual correction by humans will be always necessary in order to receive a 100% syntactically correct transcript.

3.1 Experiences

Before giving you some concrete numbers, we share our observations. First, we discovered, that the automatically generated transcript quality varies depending on the German dialect spoken. We are a university in Austria, and Austrian German varies in pronunciation and words from German German. Since Austria is with 8 millions inhabitants rather small in comparison to Germany (83 million), language transcription companies focus on German German, rather than the Austrian version. Since we have also some teachers from Germany, we could easily compare the quality of the transcript, and underline this hypothesis.

Second, even though German is our main language, many teachers use English words during their lessons for international technical terms or common words such as "review". Here the automatic transcription service totally failed, hence tried to replace the English words with a German word sounding similar. However, this was not a big surprise, since the automatic transcript software works on phonetics and it expects only German words in our case.

Third, we had several lessons where teachers showed videos in their lessons. Since most of those videos where in English, the transcription software failed due to expecting German language. However, even worse, in most of those lessons the teacher talks in parallel to the video giving additional explanations or translations, which made it even impossible for a human to understand all the words. The automatic transcription software completely failed.

Abbreviations – such as DNA, ALS – also result in a complete failure for the transcript software. Again, this is no surprise, since abbreviations are neither unambiguous nor can they be usually identified by humans to 100%. Here maybe extra, manually built, expert vocabulary will help, which is supported by main transcription services, also by the by us chosen AWS Amazon Transcribe. Of course this is connected to further human labor efforts.

Furthermore, we noticed, that connected words are not transcribed correctly. This is a special phenomenon with German language, where a lot of very long words based on the connection of basic words exist. For example, "qualitymanagement" is in German one word, and not two. However, this is only a minor issue since the transcript can be still understood despite not being fully grammatically correct.

Last but not least, punctuation marks are very often not put correctly in sentences. Our assumption is that in case a speaker makes a longer break at the end of the sentence, an end of sentence point is automatically set. Even though this should be good practice in a good speech, it does not happen frequently in our lecture talks, which have somehow more live character, e.g. interrupted by questions from students.

3.2 Concrete numbers

During the manual correction process, we recorded the labor time necessary in order to achieve a syntactically correct transcript. These numbers give a rough estimation how much human labor time you have to invest:

- For 10 minutes video recorded in an empty lecture room (see above, higher transcript quality) about 26 minutes correction time are needed, which is a factor of about 1:2.6
- For a 10 min video recorded in a full lecture room (see above, lower transcript quality) about 32 minutes were needed, which corresponds to a factor of about 1:3.2
- For a 10 min video where the teacher simply reads down a script (see above, no free speaking – highest transcript quality) about 20 min are needed for manual correction, resulting in a factor of 1:2

4 Conclusions and Future Work

As the most important result of our work, we can state, even though the quality of the speech-to-text service varies significantly, doing it from scratch – without the automated transcripts, respectively – would require much more time! Because in this case you have to create also the WebVTT format with the time codes and you have no text basis to start with. Therefore, the semi-automatic process, chosen by us, turned out to be the most efficient way in terms of pricing and invested human labor efforts.

For the set-up of the entire Open Cast platform and in particular for the integration of adding semi-automatical subtitles in your existing technical workflow, we strongly recommend utilizing help from external experts. Regarding this, we made very good experiences with the non-profit organisation Elan e.V. [15], who provide professional help with the set-up and maintenance of the Open Cast platform for reasonable prices.

For all students, who do not have the language spoken in the recorded videos as their mother tongue, subtitles help to improve understanding the content. This is also true for places with high background noises (e.g. on the bus, in a restaurant) and in places where you cannot use audio because you would disturb other people. By adding subtitles, we expect to make the learning process more efficient for the students by giving them more options.

We plan to use the transcript also as a basis for scripts, which teachers (or students) may want to prepare for their lessons. Even though it still requires some work for formatting, a lot of time for writing is saved.

The transcript with time codes can be used for a frame-accurate search. Since the whole text is tagged with time codes the search can also be extended on the video content. Hence as a search result the exact position within the video can be found.

Last but not least, we plan to use the automatically generated transcripts to be translated in other languages used at our university, which is in our case English (German is the main language). The service [8], chosen by us, currently offers 32 languages. Once a speech transcript is received it can be easily translated into other languages. This will make our recorded lectures also accessible to students not capable of speaking the German language. However, this implies a manual quality assurance because you need a syntactically and semantically error-free text in order to receive reasonable results from the automatic translation software.

References

1. H. Rehatschek: "Experiences from the Introduction of an Automated Lecture Recording System", in Proceedings of the 21st International Conference on Interactive Collaborative Learning (ICL 2018) – Volume 2 of the series Advances in Intelligent Systems and Computing, book title: The Challenges of the Digital Transformation in Education, ISBN 978-3-030-11934-8, ISSN 2194-5357, https://doi.org/10.1007/978-3-030-11935-5, pp. 151 - 162, 25–28 September 2018, Kos Island, Greece.
2. H. Rehatschek: "Design and Set-up of an Automated Lecture Recording System in Medical Education", in Proceedings of the 20th International Conference on Interactive Collaborative Learning – Volume 715 of the series Advances in Intelligent Systems and Computing, ISBN 978-3-319-73209-1,, pp 15-20, 27– 29 September 2017, Budapest, Hungary.
3. H. Rehatschek, F. Matzer, C. Vajda, C. Fazekas: "Successful Embedding of Virtual Lectures in Medical Psychology Education in Order to Improve Teacher-Student Interactivity and Collaboration" in Proceedings of the 22nd International Conference on Interactive Collaborative Learning (ICL 2019), book title: The Impact of the 4th Industrial Revolution on Engineering Education, Springer Verlag, ISBN 978-3-030-40274-7, doi.org/10.1007/978-3-030-40274-7_1, pp. 3 - 15, 25–28 September 2019, Bangkok, Thailand.
4. VITAL – VIdeo porTAL of the Medical University of Graz. <March 2020 / URL: https://vital.medunigraz.at >
5. WebVTT, The Web Video Text Tracks Format, W3C Candidate Recommendation 4 April 2019. <February 2020 / URL: https://www.w3.org/TR/webvtt1/ >
6. Paella Player, the multistream player for lectures, Open Source player for displaying two video streams synchronically. <April 2020 / https://paellaplayer.upv.es/ >
7. Auphonic, automatic audio post production web service for podcasts, broadcasters, radio shows, movies, screencasts and more. <February 2020 / https://auphonic.com/ >.
8. AWS – Amazon Transcribe, automatically convert speech to text. <February 2020 / https://aws.amazon.com/transcribe/ >
9. VTT Creator, free open source online WebVTT editor by Robert Allsopp, Arvada, Colorado. <April 2020 / https://www.vtt-creator.com>
10. YouTube, video management and streaming platform by Google. <April 2020 / https://www.youtube.com>
11. Aegisub, cross-platform open source tool for creating and modifying subtitles. <April 2020 / http://www.aegisub.org/>
12. DivXLand, <April 2020 / https://www.divxland.org/en/media-subtitler>
13. Epiphan, capture stream record. <April 2020 / URL: https://www.epiphan.com/ >
14. Opencast, open source solution for automated video capture and distribution at scale. <April 2020 / URL: http://www.opencast.org >
15. Elan eV – eLearning academic network. <April 2020 / https://elan-ev.de/>
16. Open lecture. Video portal of university of Halle. <April 2020 / http://openlecture.uni-halle.de/>
17. LTI – Learning Tools Interoperability, IMS standard. <April 2020 / https://www.imsglobal.org/activity/learning-tools-interoperability>