THE HOCHSCHUL-ASSISTENZ-SYSTEM HANS: AN ML-BASED LEARNING EXPERIENCE PLATFORM

Thomas Ranzenberger[1], Tobias Bocklet[1], Steffen Freisinger[2], Lia Frischholz[3], Munir Georges[2], Kevin Glocker[2], Aaricia Herygers[2], René Peinl[3], Korbinian Riedhammer[1], Fabian Schneider[1], Christopher Simic[1], Khabbab Zakaria[2*]
[1]*Technische Hochschule Nürnberg,*
[2]*Technische Hochschule Ingolstadt,* [3]*Hochschule Hof*
[1]*firstname.lastname@th-nuernberg.de*, [2]*firstname.lastname@thi.de*,
lia.frischholz.2@hof-university.de, *rene.peinl@hof-university.de*

BACKGROUND: The usage of e-learning platforms, online lectures and online meetings for academic teaching increased during the Covid-19 pandemic. Lecturers created video lectures, screencasts, or audio podcasts for online learning. The Hochschul-Assistenz-System (HAnS) is a learning experience platform that uses machine learning (ML) methods to support students and lecturers in the online learning and teaching processes. HAnS is being developed in multiple iterations as an agile open-source collaborative project supported by multiple universities and partners. This paper presents the current state of the development of HAnS on German video lectures.

METHOD: The current prototype of HAnS is being developed on GitHub and will be made publicly available with an open-source license [1]. HAnS is a web service consisting of multiple components. A backend contains the needed databases and engines to run the HAnS service. The HAnS application programming interface (API) handles the login, provides the data to the frontend and, in the future, to different e-learning platforms. The goal of HAnS is to use ML to process the lecture material hosted on common German e-learning platforms in order to support teaching and learning.
HAnS supports different use cases for students and lecturers. In the current prototype the students can use HAnS to search for terms tagged on the specific lecture video metadata or search a specific video for a spoken term and skip to the exact position of the spoken term (see Figure 1). The transcript is automatically created in the ML backend during the processing of the lecture material.
The ML backend uses Apache Airflow [2] to define a direct acyclic graph (DAG) in Python for processing the lecture videos. The "hans_v1.0.0" DAG consists of multiple task groups and corresponding sub-tasks (see Figure 2). The first task group downloads the uploaded media file from the backend. The media file is converted to a 16 kHz audio file in the "media_converter_audio" task group. In the next task group, the audio file is processed by an automatic speech recognition (ASR) engine. HAnS uses Kaldi-trained German acoustic and language models [3, 5] in conjunction with the Mod9 ASR Engine [4]. The final transcript is created by a natural language processing (NLP) task group which uses a multilingual transformer model to predict punctuation [6]. The search data task group uses the ASR result to create the metadata for the search engine which runs on the backend. The video data is processed to stream multiple resolutions and bitrates to the frontend video player.
The DAG is triggered by a request of the frontend and allows the configuration of different ASR or NLP engines, different video resolutions, or to skip steps such as video processing for streaming if they are not necessary. The next task group publishes the resulting artefacts including the transcript and the search metadata on the HAnS backend. Finally, the artefacts are archived in order to create a lecture course package for installation on a new instance of a HAnS backend using the HAnS package DAG.

OUTLOOK: We plan to improve and extend our prototype for our next development cycle. One focus will be to improve the ASR to handle out of vocabulary words and recognize technical terms. The second focus will be to support the search for automatically detected topics in the videos. For this, we will implement NLP components to detect corresponding topics and sub-topics. Subsequently, the frontend will be adapted for students to search for and select these topics within the video page. The evaluation of the current HAnS prototype is still ongoing. We thus expect more feature requests and suggestions for improvement from our partners as well as from the show and tell session.

[*] Co-authors listed in alphabetical order

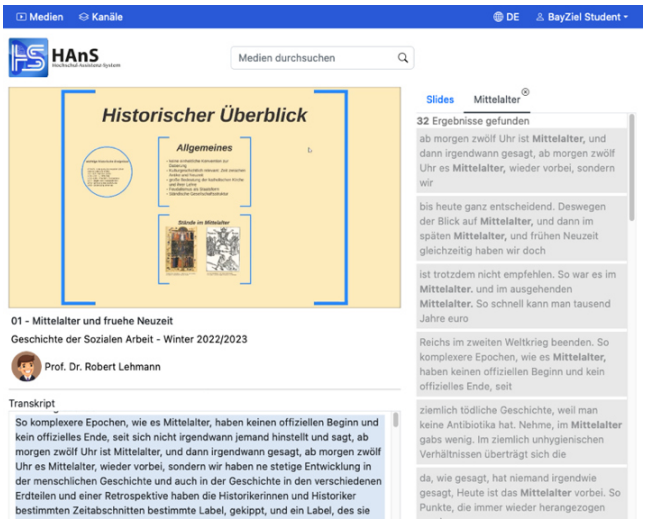

Figure 1 – Frontend video page with search results

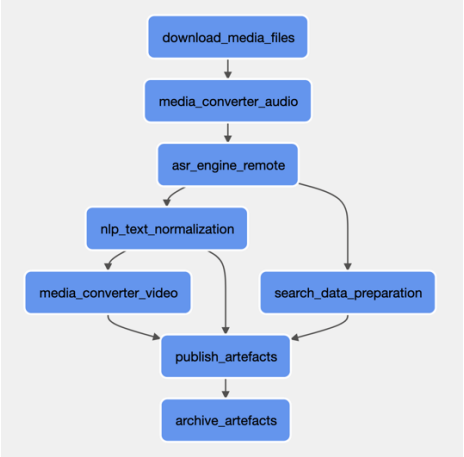

Figure 2 – HAnS Apache Airflow DAG

List of References

[1] https://github.com/th-nuernberg/hans (last accessed 13.01.2023).

[2] https://airflow.apache.org (last accessed 13.01.2023).

[3] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukáš Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlíček, Yanmin Qian, Petr Schwarz, Jan Silovský, Georg Stemmer, and Karel Vesel. 2011. The Kaldi speech recognition toolkit. *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding* (January 2011).

[4] https://mod9.io (last accessed 13.01.2023).

[5] Robert Geislinger, Benjamin Milde, and Chris Biemann. 2022. Improved Open Source Automatic Subtitling for Lecture Videos. In *Proceedings of the 18th Conference on Natural Language Processing (KONVENS 2022)*, KONVENS 2022 Organizers, Potsdam, Germany, 98–103.

[6] Oliver Guhr, Anne-Kathrin Schumann, Frank Bahrmann, and Hans Joachim Böhme. 2021. FullStop: Multilingual Deep Models for Punctuation Prediction. *Proceedings of the Swiss Text Analytics Conference 2021* (June 2021).