# EXTENDING HANS: LARGE LANGUAGE MODELS FOR QUESTION ANSWERING, SUMMARIZATION, AND TOPIC SEGMENTATION IN AN ML-BASED LEARNING EXPERIENCE PLATFORM

*Thomas Ranzenberger*[1]*, Tobias Bocklet*[1]*, Steffen Freisinger*[1]*, Munir Georges*[2]*,*
*Kevin Glocker*[2]*, Aaricia Herygers*[2]*, Korbinian Riedhammer*[1]*, Fabian Schneider*[1]*,*
*Christopher Simic*[1]*, Khabbab Zakaria*[2] *

[1]*Technische Hochschule Nürnberg,* [2]*Technische Hochschule Ingolstadt*
[1]*firstname.lastname@th-nuernberg.de,* [2]*firstname.lastname@thi.de*

**Abstract:** The use of chatbots based on large language models (LLMs) and their impact on society are influencing our learning experience platform Hochschul-Assistenz-System (HAnS). HAnS uses machine learning (ML) methods to support students and lecturers in the online learning and teaching processes [1]. This paper introduces LLM-based features available in HAnS which are using the transcript of our improved Automatic Speech Recognition (ASR) pipeline with an average transcription duration of 45 seconds and an average word error rate (WER) of 6.66% on over 8 hours of audio data of 7 lecture videos. A LLM-based chatbot could be used to answer questions on the lecture content as the ASR transcript is provided as context. The summarization and topic segmentation uses the LLM to improve our learning experience platform. We generate multiple choice questions using the LLM and the ASR transcript as context during playback in a period of 3 minutes and display them in the HAnS frontend.

## 1 Method

The current prototype of HAnS supports different use cases for students and lecturers [1]. We added additional tasks to the Apache Airflow direct acyclic graph (DAG) to generate short summaries, long summaries and perform topic segmentation on the video's closed captions file using a finetuned Llama2-based Vicuna-13b v1.5 model with 16000 tokens input length including context [2]. We use the transcript and closed captions file generated by our ASR task group in the Airflow DAG to extend the LLM prompt context for implementing our LLM-based features.

### 1.1 ASR Performance Evaluation

The closed captions file is generated by an ASR task in the DAG. In previous HAnS versions we used a Kaldi-based recognizer with a German speech model [3], [4]. Our current system is based on OpenAi's Whisper (Whisper) [5]. This gives us the advantage to support German and English lectures and generate transcripts and subtitles for HAnS. An additional requirement of HAnS are word level timestamps. HAnS needs the timestamps to skip to specific spoken words in the HAnS video player. In order to deploy HAnS for an on-the-fly upload scenario we need to reduce the transcription time and find a trade-off between the time and the transcription

*Co-authors listed in alphabetical order

DOI: 10.35096/othr/pub-7103, URN: urn:nbn:de:bvb:898-opus4-71032

accuracy for our use case. Therefore we analysed different approaches provided by software libraries that speed-up the transcription time of Whisper.

The long form audio transcription test task for different software libraries was conducted on a German lecture video with a duration of 1 hour 4 minutes, and 15 seconds. For the last semester period we used the Whisper medium model, which we use now for our comparison baseline. All models use 16 bit floating point precision and a single A100 GPU with 40GB VRAM. We compare the python packages openai-whisper [6], insanely-fast-whisper [7], and whisper-s2t [8] and the whisper.cpp [9] implementation to compare WER and transcription time in Table 1. The used beam size for all libraries is 5 except insanely-fast-whisper with a beam size of 1. The evaluation used german-asr-lm-tools [10] for text normalization and punctuation removal and JiWER [11] for calculating the WER on lower case reference and hypothesis texts. The reference transcription was provided by a professional transcription service.

Table 1 shows that the usage of FlashAttention-2 [12] of insanely-fast-whisper results in a huge improvement in transcription time compared to openai-whisper with an average speed-up factor of approx. 10 and whisper.cpp with a speed-up factor of approx. 5. The usage of whisper-s2t and the underlying CTranslate2 inference engine optimizations [13] speed-up the transcription time towards our required execution times for HAnS. Depending on the used GPU a higher batch size of 48 could improve the transcription duration. The fastest transcription was performed by using whisper-s2t with a large-v2 based model. The used GPU memory increases depending on the batch size. GPU memory peaks reach 24 GB VRAM with batch size 48, batch size 16 consumes 10 GB VRAM for our long transcription task. The best WER on the single German video lecture audio was 7.85% using whisper.cpp with a large-v3 based model.

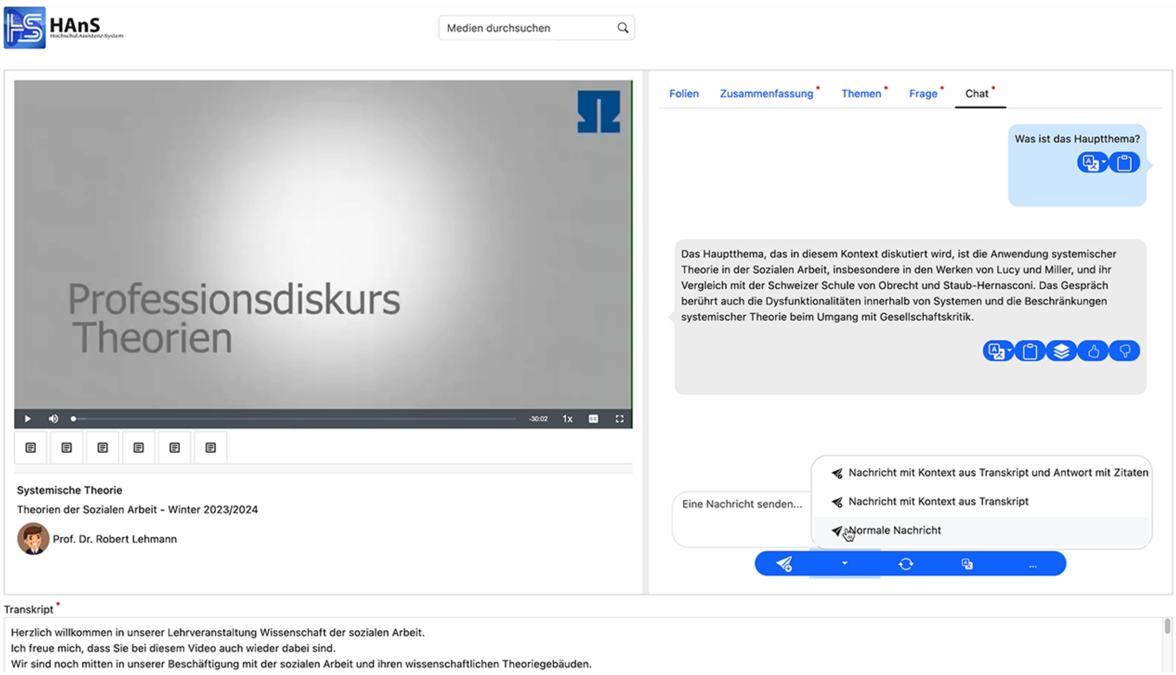
An additional evaluation on the complete German lecture with 7 videos with large-v3 and whisper-s2t took 05:15 to transcribe 8 hours 14 minutes, and 24 seconds of long form audio data. The reference transcripts were generated by the same professional transcription service and we used the same evaluation approach as the previous experiment. The average transcription time was 45 seconds. The lowest WER was 5.23% and the highest 7.94%. The average WER was 6.66% and a comparison between large-v2 and large-v3 showed a maximum WER improvement of 0.76% for large-v3 over large-v2 for one single lecture video. We found the best trade-off is to use whisper-s2t for providing a fast and accurate transcript for the user and our in the following described LLM tasks within Airflow DAG and HAnS web frontend.

**Table 1** – Comparison of different python packages and whisper.cpp using Whisper models with different batch size and resulting word error rates, and transcription times in minutes and seconds.

| Package | Model | WER | Duration | Batch Size |
|---|---|---|---|---|
| openai-whisper | medium | 11.09% | 08:08 | Default |
| openai-whisper | large-v3 | 10.34% | 14:12 | Default |
| whisper.cpp | ggml-large-v3 | 7.85% | 05:29 | Default |
| insanely-fast-whisper | large-v3 | 10.45% | 01:21 | 24 |
| whisper-s2t | Systran/faster-whisper-large-v3 | 7.94% | 00:46 | 16 |
| whisper-s2t | Systran/faster-whisper-large-v3 | 7.94% | 00:45 | 24 |
| whisper-s2t | Systran/faster-whisper-large-v3 | 7.94% | 00:40 | 48 |
| whisper-s2t | Systran/faster-whisper-large-v2 | 8.08% | 00:29 | 48 |

### 1.2 LLM-based Features with ASR Context

In order to display short summaries on the HAnS frontend view of the search results together with the words found in the transcript of each lecture video we use the transcript of the ASR



Systemische Theorie

Theorien der Sozialen Arbeit - Winter 2023/2024

Prof. Dr. Robert Lehmann

Was ist das Hauptthema?

Das Hauptthema, das in diesem Kontext diskutiert wird, ist die Anwendung systemischer Theorie in der Sozialen Arbeit, insbesondere in den Werken von Lucy und Miller, und ihr Vergleich mit der Schweizer Schule von Obrecht und Staub-Hernasconi. Das Gespräch berührt auch die Dysfunktionalitäten innerhalb von Systemen und die Beschränkungen systemischer Theorie beim Umgang mit Gesellschaftskritik.

Transkript

Herzlich willkommen in unserer Lehrveranstaltung Wissenschaft der sozialen Arbeit.
Ich freue mich, dass Sie bei diesem Video auch wieder dabei sind.
Wir sind noch mitten in unserer Beschäftigung mit der sozialen Arbeit und ihren wissenschaftlichen Theoriegebäuden.

**Figure 1** – Frontend video page with chatbot and prompt modes

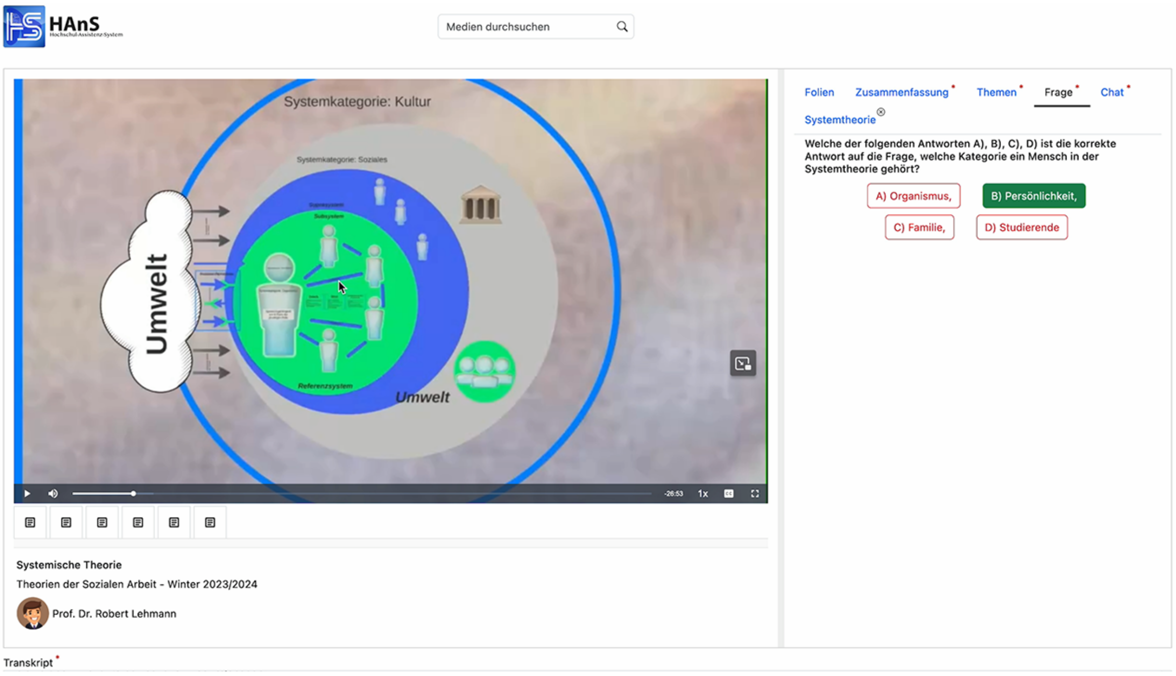
system for the LLM prompt context. The generated short summary enables the students and lecturers to decide on the most relevant lecture video which corresponds to the search input.

The generated long summaries are displayed in the video player view in an own workbench tab. LLM segmented topics together with corresponding timestamps for navigating in the current lecture video are displayed in an additional workbench tab and in a shortcut bar. This allows users to skip to specific topic segments within the lecture video directly.

To answer questions related to the lecture video, we added an LLM-based chatbot to the HAnS frontend (see Figure 1). We introduce different prompt modes to chat with the LLM. The first mode uses only the LLM knowledge without any additional input from the ASR transcript of the video lecture. The second mode uses the transcript as context and sends it to a retrieval service using a Generative T5 Retriever model [14] which decides for the most relevant parts of the transcripts to answer or fulfill the user prompt.

The retrieval service returns multiple sections used for the final prompt request to the LLM. The third mode uses the same approach as the second mode but adds additional instructions to the prompt for citing the context sections. We instruct the LLM to write the number of the section provided by the context on the specific section of the LLMs answer. The different modes provide the possibility to explore if a question could be already answered by the LLM without the lecture context, if the answer improves with additional context from the lecture transcript, and if the LLM cites the provided context sections correctly. The context could be shown using the stack button on a specific LLM answer and each citation included in the answer displays the used context section. This enables lecturers and students to verify the LLM answers with the transcript as ground truth.

As our lectures are conducted in German and English, we added an additional translation option to translate the users chat message to English and translate the LLM answer back to German. This enables the students and lecturers to ask the same question in both languages and verify and compare the answers for these two languages. The translation service is based on Facebook FAIR's WMT19 submission [15]. If the translate option was used the user can display the original and translated messages in the chatbot tab by selecting the translation button on the corresponding messages. The translation option is only valid for German locale. If the English locale is selected in the frontend the translation is not necessary and therefore disabled.



Welche der folgenden Antworten A), B), C), D) ist die korrekte Antwort auf die Frage, welche Kategorie ein Mensch in der Systemtheorie gehört?

A) Organismus,

B) Persönlichkeit,

C) Familie,

D) Studierende

Systemische Theorie

Theorien der Sozialen Arbeit - Winter 2023/2024

Prof. Dr. Robert Lehmann

**Figure 2** – Frontend video page with LLM generated multiple choice questions

The last testable LLM feature is generating multiple-choice questions and 4 answer options of the lecture video every 3 minutes (see Figure 2). The questions are generated using the text of the transcript section corresponding to the 3 minutes period as context for the prompt. The parsed question and answer choices of the LLM are displayed on a separate workbench tab in the video player view. This allows students to interactively check their current state of knowledge. Lecturers could use the LLM-based chatbot to gather inspiration for exam tasks and ask questions regarding the lecture and further explore the capabilities of the LLM for preparing lectures or creating learning objective tests.

## 2 Outlook

We plan to explore, improve, and conduct studies on the LLM-based features together with our partners for our next development cycle. As the LLMs quality, context size, and support for multiple modalities evolves, we expect to further extend and evaluate our learning experience platform HAnS.

## References

[1] Ranzenberger, T., T. Bocklet, S. Freisinger, L. Frischholz, M. Georges, K. Glocker, A. Herygers, R. Peinl, K. Riedhammer, F. Schneider, C. Simic, and K. Zakaria: *The Hochschul-Assistenz-System Hans: An ML-based Learning Experience Platform.* In C. Draxler (ed.), *Studientexte zur Sprachkommunikation: Elektronische Sprachsignalverarbeitung 2023*, pp. 168–169. TUDpress, Dresden, 2023. URL `https://www.essv.de/pdf/pdf/2023_168_169.pdf`.

[2] Zheng, L., W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. Xing, H. Zhang, J. E. Gonzalez, and I. Stoica: *Judging LLM-as-a-judge with MT-Bench and Chatbot Arena.* In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*. 2023. URL `https://openreview.net/forum?id=uccHPGDlao`.

[3] Milde, B. and A. Köhn: *Open Source Automatic Speech Recognition for German.* In *Proceedings of ITG 2018*, pp. 251–255. Oldenburg, Germany, 2018.

[4] Geislinger, R., B. Milde, and C. Biemann: *Improved Open Source Automatic Subtitling for Lecture Videos.* In *Proceedings of the 18th Conference on Natural Language Processing (KONVENS 2022)*, pp. 98–103. KONVENS 2022 Organizers, Potsdam, Germany, 2022. URL `https://aclanthology.org/2022.konvens-1.11`.

[5] Radford, A., J. W. Kim, T. Xu, G. Brockman, C. Mcleavey, and I. Sutskever: *Robust Speech Recognition via Large-Scale Weak Supervision.* In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, vol. 202 of *Proceedings of Machine Learning Research*, pp. 28492–28518. PMLR, 2023. URL `https://proceedings.mlr.press/v202/radford23a.html`.

[6] *openai/whisper: Robust Speech Recognition via Large-Scale Weak Supervision.* `https://github.com/openai/whisper/commit/e58f288`, 2022. Accessed: 2024-01-26.

[7] *Vaibhavs10/insanely-fast-whisper: An opinionated CLI to transcribe Audio files w/ Whisper on-device! Powered by Transformers, Optimum and flash-attn.* `https://github.com/Vaibhavs10/insanely-fast-whisper/commit/ff0df40`, 2023. Accessed: 2024-01-26.

[8] *shashikg/WhisperS2T: An Optimized Speech-to-Text Pipeline for the Whisper Model Supporting Multiple Inference Engine.* `https://github.com/shashikg/WhisperS2T/commit/996f2f0`, 2023. Accessed: 2024-01-26.

[9] *ggerganov/whisper.cpp: Port of OpenAI's Whisper model in C/C++.* `https://github.com/ggerganov/whisper.cpp/commit/0b9af32`, 2022. Accessed: 2024-01-26.

[10] *bmilde/german-asr-lm-tools: Crawling and creating a German language model resource.* `https://github.com/bmilde/german-asr-lm-tools`, 2020. Accessed: 2024-01-26.

[11] Morris, A., V. Maier, and P. Green: *From WER and RIL to MER and WIL: improved evaluation measures for connected speech recognition.* 2004. doi:10.21437/Interspeech.2004-668.

[12] DAO, T.: *FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning*. 2023. 2307.08691.

[13] *OpenNMT/CTranslate2: Fast inference engine for Transformer models*. https://github.com/OpenNMT/CTranslate2, 2019. Accessed: 2024-01-26.

[14] NI, J., C. QU, J. LU, Z. DAI, G. HERNANDEZ ABREGO, J. MA, V. ZHAO, Y. LUAN, K. HALL, M.-W. CHANG, and Y. YANG: *Large Dual Encoders Are Generalizable Retrievers*. In Y. GOLDBERG, Z. KOZAREVA, and Y. ZHANG (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 9844–9855. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2022. doi:10.18653/v1/2022.emnlp-main.669. URL https://aclanthology.org/2022.emnlp-main.669.

[15] NG, N., K. YEE, A. BAEVSKI, M. OTT, M. AULI, and S. EDUNOV: *Facebook FAIR's WMT19 News Translation Task Submission*. In O. BOJAR, R. CHATTERJEE, C. FEDERMANN, M. FISHEL, Y. GRAHAM, B. HADDOW, M. HUCK, A. J. YEPES, P. KOEHN, A. MARTINS, C. MONZ, M. NEGRI, A. NÉVÉOL, M. NEVES, M. POST, M. TURCHI, and K. VERSPOOR (eds.), *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pp. 314–319. Association for Computational Linguistics, Florence, Italy, 2019. doi:10.18653/v1/W19-5333. URL https://aclanthology.org/W19-5333.