INTERSPEECH 2021: Show & Tell Contribution
30 August – 3 September, 2021, Brno, Czechia

Live Subtitling for BigBlueButton with Open-Source Software

Robert Geislinger[1,2], *Benjamin Milde*[1,2], *Timo Baumann*[2], *Chris Biemann*[2]

[1]Hamburger Informatik Technologie-Center e.V., Germany
[2]Language Technology Group, Universität Hamburg, Germany
6geislin,milde,baumann,biemann@informatik.uni-hamburg.de

Abstract

We present an open source plugin for live subtitling in the popular open source video conferencing software BigBlueButton. Our plugin decodes each speaker's audio stream separately and in parallel, thereby obliviating the need for speaker diarization and seamlessly handling overlapped talk. Any Kaldi-compatible nnet3 model can be used with our plugin and we demonstrate it using freely available TDNN-HMM-based ASR models for English and German. Our subtitles can be used as they are (e.g., in loud environments) or can form the basis for further NLP processes. Our tool can also simplify the collection of remotely recorded multi-party dialogue corpora.

Index Terms: Automatic speech recognition, videoconferencing, meeting transcription, computer-supported collaborative work, automatic subtitles, multi-party dialogue, VoIP

1. Introduction

In the COVID-19 pandemic, the opportunity for professional and private physical meetings is severely limited. Professional and academic events are being transformed from face-to-face to digital events. The BigBlueButton (BBB) open source conferencing system for online meetings and digital classrooms[1] has become increasingly popular (as have some commercial alternatives). BBB can be used from the web browser of most devices with no additional software needed. BBB allows to create breakout rooms, upload slides and work collaborative on a whiteboard.

Subtitles display the spoken content of a video in written text. Primarily, subtitles support accessibility of speech for persons with hearing limitations. Beyond this purpose, they can also be helpful at tasks like consume video in silence (or in noisy environments), or as a basis for further text-based tasks such as search, or summarization, or tranlation to assist non-native speakers. Conventionally, subtitles are produced offline and this has become more relevant during the pandemic, e.g. to improve the accessibility of e-lectures [1]. While subtitles can be created manually in BBB, this is a very time consuming process. Automatic Speech Recognition (ASR) helps to provide live subtitles in a high quality at low cost and can also be computed on the fly.

In this paper, we present a Kaldi-based [2] plugin to add automatically generated subtitles into BBB conferences and breakout rooms by using open-source software and pre-trained models without any licensing cost or advanced knowledge in ASR is needed. Our plugin is freely available[2] and supports German and English out of the box.

[1]https://bigbluebutton.org/
[2]https://github.com/uhh-lt/bbb-live-subtitles

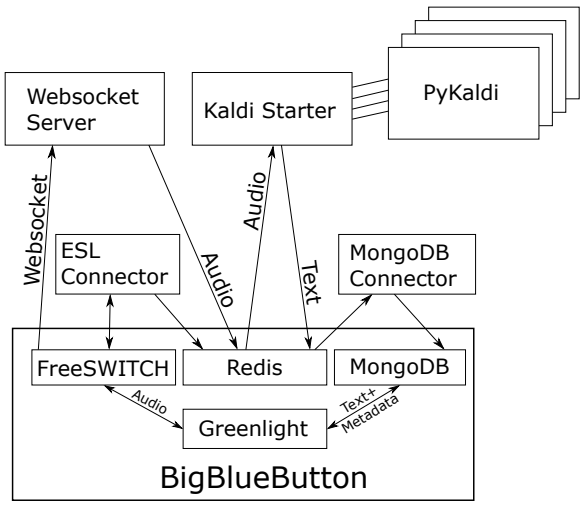

Figure 1: *Schematic diagram of the system architecture.*

2. Architecture

The software is designed with a distributed micro-service architecture and written primarily in Python. The architecture design makes the deployment of our plugin very flexible; the components can be either installed on the same server that hosts BBB or distributed to remote machines depending on the computing resources. See Figure 1 for an overview of all components and how they are related to BBB's software stack.

The communication of the services among each other is realised through the BBB provided Redis database. BBB uses FreeSWITCH[3] to route audio and video signals from and to the participants of a conference. BBB's components and services communicate through the FreeSWITCH API event socket library (ESL), Redis database and MongoDB database. To install and run our plugin, only minimal changes need to be made to an existing BBB system configuration. We use the FreeSWITCH module *audio fork by drachtio*[4] to record the audio signal of each participant separately and stream it into the ASR processing pipeline via websockets. When a participant enters a conference or breakout room the module starts to stream the incoming audio. It is then copied into a Redis pubsub channel, that is subscribed and consumed by the ASR services.

For ASR we use the Kaldi toolkit [2] through PyKaldi [3] with pre-trained models for German [4] and English [5]. For every new participant, a new kaldi-model-server[5] is started and

[3]https://freeswitch.com/
[4]https://github.com/drachtio/drachtio-freeswitch-modules/tree/master/modules/mod_audio_fork
[5]https://github.com/uhh-lt/kaldi-model-server

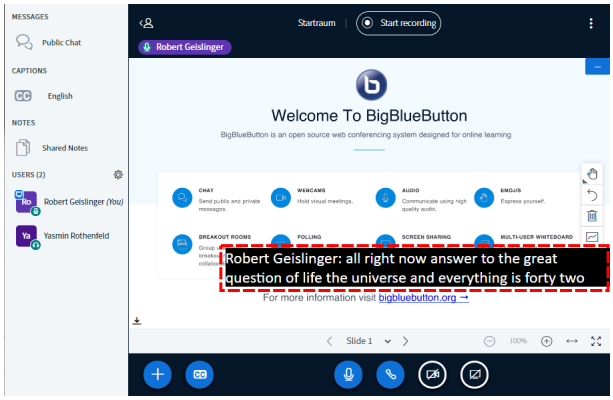

Figure 2: *Screenshot of a BBB conference in Greenlight. Generated subtitles (red box) are prefixed with the speaker's name.*

connected to the channel which then loads the ASR model with PyKaldi and connects to the Redis channels and handles online decoding. The generated transcription is amended with speaker tags and published into the BBB provided MongoDB, which is scanned by the web frontend that updates partial and completed transcriptions several times per second.

3. Web Interface

Greenlight is the standard web interface for BBB. It is mainly written as a HTML5 application and served via Ruby on Rails and can be used with most modern browsers regardless of whether on mobile phone or a PC.

The subtitle functionality for displaying subtitles in Greenlight is also used by our tool. Instead of manually written, they are created automatically on the server by processing the participants' audio streams. Participants can individually decide whether to display the provided subtitles or not, and they can also be activated or deactivated for the entire conference by the presenter. See Figure 2 for an example screenshot of the generated subtitles displayed in a BBB conference.

4. Evaluation

We evaluate the effects of the integration of Kaldi with BigBlueButton in terms of WER degradation. WER may be influenced by online decoding (rather than speaker-specific multi-pass decoding in offline decoding), effects of encoding/decoding audio with the Opus codec [6] using BBB's standard settings, and potential packet loss when using VoIP transmission from the Greenlight client to FreeSWITCH. We here evaluate the German ASR performance on the Verbmobil [7] corpus, a challenging German ASR test set for spontaneous speech. We expect similar influences for other languages. See Table 1 for the WER results obtained. Note that ASR results can often be meaningful to humans even at high absolute WER.

As can be seen in the table, all three processing limitations yield a performance penalty. Online decoding increases WER by 6–12 % relative. Passing audio through the Opus codec (same quality settings as in BBB), yields a small penalty of 1–8 % relative due to lossy compression and audio artifacts. Finally, we test VoIP effects (packet loss or latency beyond what is acceptable for real-time settings in FreeSWITCH) by streaming audio with a virtual microphone from Greenlight into BBB using a web browser and remote server, as a participant would use our pipeline. VoIP-related effects incur a penalty of 3–4 %.

Table 1: *Influence on WER of ASR limitations in the BBB setup compared to speaker-dependent offline decoding with Kaldi (Verbmobil German corpus: spontaneous speech, open-ended vocabulary).*

		WER in %	
Setup	**Limitation**	**VM1**	**VM2**
Plain Kaldi	—	27.4	30.4
PyKaldi	online decoding	30.7	32.2
+ Opus codec	audio artifacts	31.0	34.7
+ via BBB	packet loss	32.3	35.8

In total, the pipeline incurs a penalty of 5–11 % over direct online decoding and 18 % over what can be achieved with batch decoding in an offline setting.

5. Conclusion

We integrated a fully open-source automatic speech recognition system into the existing open-source conferencing software BigBlueButton. The Opus codec and VoIP packet loss as found in Greenlight/BBB slightly reduce WER performance obtained with off-the-shelf ASR models using online decoding. However the observed effect is modest with a WER degradation of 5–11 % relative.

Instructions to download freely available pre-trained ASR models for German and English are also provided, integration of custom models or Kaldi nnet3 models for other languages is straightforward.

Our solution can be used as a foundation for further integration of NLP processes (summarization, search, . . .) into BigBlueButton. Further improvements can be achieved by optimizing our decoders for the respective single user, e.g. by storing long-term speaker adaptation states with each user ID.

6. References

[1] B. Milde, R. Geislinger, I. Lindt, and T. Baumann, "Open source automatic lecture subtitling," in *Proceedings of ESSV 2021*, Virtual Berlin, Germany, 2021, pp. 128–134.

[2] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi speech recognition toolkit," in *ASRU Workshop*, Waikoloa, USA, 2011.

[3] D. Can, V. R. Martinez, P. Papadopoulos, and S. S. Narayanan, "PyKaldi: A python wrapper for Kaldi," in *Int. Conf. on Acoustics, Speech and Signal Processing*, Calgary, Canada, 2018, pp. 5889–5893.

[4] B. Milde and A. Köhn, "Open source automatic speech recognition for German," in *Proc. of ITG*. Oldenburg, Germany: VDE, 2018, pp. 251–255.

[5] F. Hernandez, V. Nguyen, S. Ghannay, N. Tomashenko, and Y. Estève, "TED-LIUM 3: twice as much data and corpus repartition for experiments on speaker adaptation," in *Int. Conf. on Speech and Computer (SPECOM)*, Leipzig, Germany, 2018, pp. 198–208.

[6] J.-M. Valin, K. Vos, and T. Terriberry, "Definition of the Opus audio codec," RFC 6716, Sep. 2012.

[7] W. J. Hess, K. J. Kohler, and H.-G. Tillmann, "The Phondat-Verbmobil speech corpus," in *Fourth European Conf. on Speech Communication and Technology*, Madrid, Spain, 1995, pp. 863–866.