Aurora Echo – Audio-visual Service Feedback System

Ethan Goh
7086cmd@gmail.com
Zhenhai High School *

Bingzhen Wu
Zhenhai High School †

Qinyu Wang
Zhenhai High School ‡

Pengzhi Chu
Shanghai Jiao Tong University

Xiaoni Liang
Shanghai Jiao Tong University

Abstract

Service feedback systems are widely used in daily life, but the traditional text feedback system is not vivid and direct enough. Users will only give its rate and sometimes write a few words, or be required to fill in a blank with staggering amount of questions, which is not enough for the resource provider to improve its service. In this paper, we propose a new feedback system, named **Aurora Echo**, which provides audio-visual feedback for the resource provider. Through numerous artificial intelligence, such as LLM, we can make the system more friendly to not only user but also the resource provider.

*The article is written during the summer camp held by Shanghai Jiao Tong University, which provides us a stage to crate innoative applications with artificial intelligence. Ethan Goh's Chinese Pinyin name: Wu Chengyu. Ethan Goh designed the project and the architecture of the system, and takes the responsibility of the development.

†Wu develops some extensions for the system, and takes the responsibility of the testing.

‡Wang makes the presentation for the project, and participate in the development.

Preprint. Under review.

1 Introduction

In daily life, resource provider may want users to provide the feedback of the resource[4], but in text form, it is hard to capture the direct response, including the facial emotion, the sense of speaking. In text form, we sometimes rate it as 5 stars, but in fact we are not satisfied with the service, since we just want to get the reward, or we are too lazy to write the feedback. It is also boring to type comments, especially facing the "50 words at least" and similar requirement.

Is there a way to provide feedback in a more vivid and direct way? Associating the video conference, we adapted the form and integrate artificial intelligence algorithms to provide a more vivid and direct feedback system.

The **Aurora Echo** system, as we named, is a system that provides audio-visual feedback for the resource provider. It mainly uses `transformers`[28] provided by HuggingFace, `PyTorch`[20] provided by Meta, and `MediaPipe`[16] provided by Google, to recognize the facial emotion, the sense of speaking, and the content of the speech. Then we can use Large Language Models (LLMs), such as Llama-3, to summarize and analyze the feedback, giving effective response to the resource provider and the feedback giver.

The innovative parts of the project are:

- The system uses the audio-visual way (inspired by video conference) to fetch the feedback, which is more vivid and direct.
- The system introduces Large Language Models (LLMs) to summarize and analyze the feedback. We can use LLMs to generate the response to the IS-friendly format, to statistic the feedback, and help the resource provider to improve its service.
- Privacy is considered in the system. We apply mosaic to the video feedback, and we do not store the video feedback, only the audio feedback and the analysis.
- The interface for frontend. We provide a web interface for not only the resource provider to check the feedback and the analysis, but also the feedback giver to feedback the response and track the feedback.

2 Background

The traditional feedback system experienced a long history. From the users, the feedback is the most direct way for resource providers to know how the users feel about the resource.[11]

First, people use suggestion boxes to collect feedback. It can be regarded as the first step of the feedback system. Users are required to write down their feedback on a piece of paper and put it into the suggestion box, which takes a long time to collect and analyze the feedback.

Then, companies may contact customers to have a survey or an interview. It makes the process of feedback more direct, but it still requires a lot of time and human resources. With the proliferation of cell phones, companies may employ the SMS or email to collect feedback, which is more efficient than the traditional way.

Nowadays, companies may ask users to rate the service in several seconds, or provide a form to fill in. It is friendlier to the automation system, but either user's time or the completeness of the feedback is not enough.

Above all, the traditional feedback system requires a lot of human resources, and the feedback is not vivid and direct enough. With the development of LLMs, people may use Large Language Model to analyze the text feedback, but still, the most immediate aspect cannot be represented.

3 Architecture

The figure 1 shows the basic architecture of the **Aurora Echo** system. The **Aurora Echo** system is mainly composed of three parts: the audio process system, the visual recognition system, and the feedback analysis system. Beside of the core system, we also provide a web interface for the resource provider to check the feedback and the analysis.

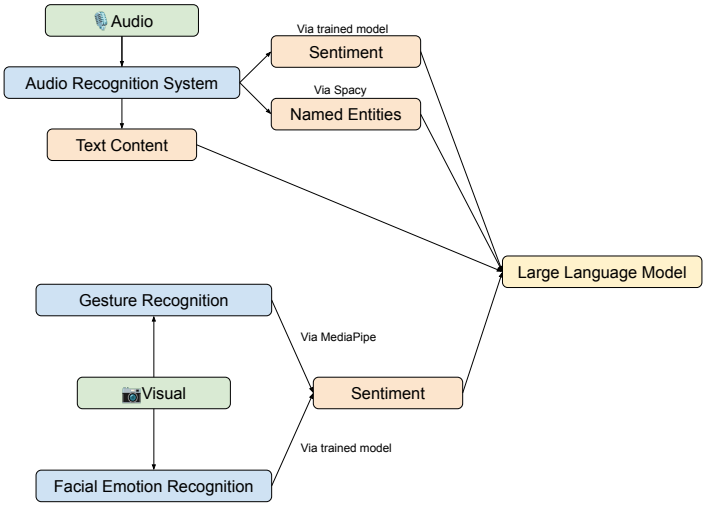

Figure 1: The basic architecture of the **Aurora Echo** system.

3.1 Audio Process System

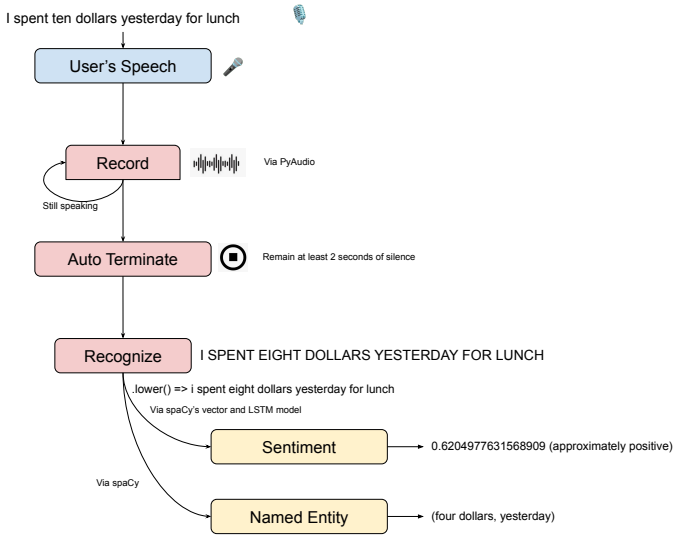

Figure 2: The audio process system of **Aurora Echo**.

The Audio Process System of **Aurora Echo** is responsible for recording the audio feedback and converting it into text. We also require the system to initially recognize the named entity and the sentiment of the speech.

3.1.1 Audio Recording

Through `PyAudio`[27] library, we can record the audio feedback, and stop recording after 2 s of silence.

According to the formula of calculating RMS level of the signal in dB (formula 1), we can calculate the RMS level of the audio feedback.

$$\text{RMS} = 20 \times \log_{10} \left(\frac{1}{N} \sum_{i=1}^{N} x_i^2 \right) \tag{1}$$

The algorithm of the stop recording is, when the RMS level of the audio feedback is less than 30 dB, we stop recording.

3.1.2 Audio Recognition

Then we use the `wav2vec` model [6], we can convert the audio into text. According to the configuration of the pilot test environment, we selected the `wav2vec2-large-960h` model, which is a pre-trained model provided by HuggingFace and Facebook.

The result of the audio recognition is a text with all capital letters, and we used `spaCy`[12, 13] library to load the sentence.

3.1.3 Sentiment Analysis

The sentiment analysis is a logistic regression model trained on the `IMDb`[19, 17] dataset. We use the LSTM[21] model to train the sentiment analysis model.

3.1.4 Named Entity Recognition

We directly use the `spaCy` library to recognize the named entity in the text.

3.2 Visual Recognition System

The visual recognition system involves the `MediaPipe` library to recognize the face position and the gesture, and a fine-tuned ResNet-34 [10] for the facial emotion recognition.

The figure 3 shows the visual recognition system of **Aurora Echo**.

3.2.1 Facial Emotion Recognition

Through the `MediaPipe` library, we can recognize the face position and the facial emotion of the feedback giver.

We normalized the image with following steps:

1. Recognize the face via `MediaPipe`.
2. Crop the face region.
3. Resize the face region to 256×256.
4. Automatically adjust the brightness, contrast, and saturation, and automatically rotate and flip the image.
5. Normalize the image.

Then, we adjust the `fc` layer of the fine-tuned ResNet-34 model to recognize the facial emotion.

The datasets are collected from `Kaggle`[24, 15], and artificially collected them into angry, disgust, fear, happy, neutral, sad, and surprise.

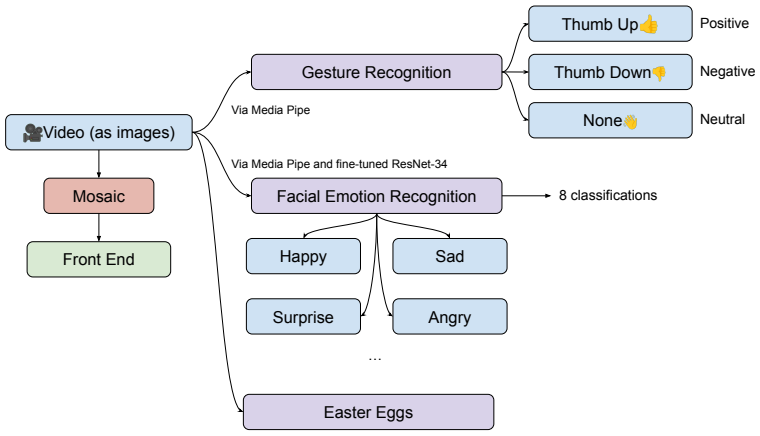

Figure 3: The visual recognition system of **Aurora Echo**.

3.2.2 Gesture Recognition

The gesture recognition is an accessibility function that helps the model knows the feedback giver's reaction more. We do not apply the ML model to the gesture recognition, but we use the `MediaPipe` library to recognize the gesture. Then we classified the gesture into 3 categories: thumbs up, thumbs down, and others.

3.3 Large Language Model Integration

The Large Language Model (LLM) is the core of the feedback analysis system.

Considering the environment, we selected `Llama-3.1-8B`[3, 26][1], or `Qwen2-1.5B`[2] as the offline LLM, or `gpt-4o-mini`[18] or `Llama-3.1-405B` as the online LLM.

The LLM is responsible for summarizing the feedback, analyzing the feedback, and generating the response.

The invocation prompt is attached in the appendix.

4 Why Aurora Echo?

Using **Aurora Echo** means that you are using a vivid and direct feedback system. It not only pursues the directness of the feedback, a deep integration of LLMs, and a privacy-guaranteed system.

4.1 Security and Privacy

The security issue of artificial intelligence is a hot topic. Calling unauthorized APIs, or storing the data without permission, is a serious problem. In **Aurora Echo**, system not only provides interfaces for calling the APIs, but also recommend users who have the ability to run the model locally.

Taking `Qwen2-1.5B` as an example, the performance requirements is lower than other models, but its usability is still guaranteed for the feedback analysis. We tested the model on Apple's M2 Pro, we can generate the response in about 3 min. It may be slow, but the performance of devices running Large Language Models are better.

[1] Actually the model always ends with `-Instruct`, but for the sake of typography, we just ignored the suffix.

We also provide API to OpenAI's ChatGPT-4o Mini. The model is a smaller version of ChatGPT-4o, which is a model that can generate the response in real-time.

4.2 Multimodal Feedback

Not only the text feedback, but also the audio-visual feedback is provided. It enhances the directness of the feedback, reflecting the most immediate aspect of the feedback giver. Though Large Language Models now are available to directly have conversations "face"-to-face, costs may be higher than this system.

5 Comparison

The ChatGPT-4o can understand the video very well, so it can also abstract the feedback from the video. [23] But why we do not use ChatGPT-4o to abstract the feedback from the video?

5.1 Cost

Just taking the image module as an example, and the audio module similar.

According to a small calculation, if you put a 30 s video to a LLM, you may spend about 200 GiB of data, and about 8 kW energy for parsing the video. [14, 22]

If using the Aurora Echo, you can just run the model on the local CPU, even on the web (with WASM and MediaPipe web), you can save a lot of energy and data.

Suppose there were a video in 30 s (30 fps). We just need to recognize 900 frames of the gesture and face. Amount parameters of models (including the facial emotion recognition) related to image processing is at most 2 Billion parameters.

Also, the extraction of LLM is also energy-consuming.

Though finally Aurora Echo uses the LLM to generate the `function call` and to summarize, this kind of task doesn't require a lot of parameters, and the model can also run on the local CPU.

5.2 Privacy

Aurora Echo runs model on the local. In the best case scenario, ALL the models, including the LLM, can run on local (with the help of uniformed memory from Apple Silicon, or using OpenVINO for Intel CPU). Through functions above, we can easily run `Qwen2-1.5B`, or `Llama-3.1-8B` on the local, once you have at lease 16 GiB of memory.

Also, the face is mosaic-permitted, and the video is not stored. What the system stores is only the report and the function-call to the database.

5.3 Directness

The directness of the feedback is the most important aspect of the system.

Through Aurora Echo, we can also promptly expand the conversation with the help of LLM. We can use prompt to invoke the LLM, and have a chat to response the most immediate aspect of the feedback giver.

6 Conclusion

The **Aurora Echo** system is a new feedback system that provides audio-visual feedback for the resource provider. Through its 3 parts, the audio process system, the visual recognition system, and the feedback analysis system, we can provide a more vivid and direct feedback system for the resource provider. The system is also privacy-friendly, and we provide a web interface for the resource provider to check the feedback and the analysis.

The code of the system is available at `https://github.com/zzteam-rccup/aurora-echo.git`.

The next step for the project are:

- Improve the frontend interface.
- Migrate the MediaPipe and the OpenCV part to the web, with the help of WASM and Rust.
- Add some extensions for the system.
- Improve the accuracy of facial emotion recognition.
- Optimize the performance of the system.

Acknowledgements We are grateful to Shanghai Jiao Tong University for providing the summer camp, and the teachers for guidance of artificial intelligence.

References

[1] *Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part I*. Springer International Publishing, 2018.

[2] Qwen2 technical report. 2024.

[3] AI@Meta. Llama 3 model card. 2024.

[4] K.J. Åström and R.M. Murray. *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press, 2010.

[5] Guy Azov, Tatiana Pelc, Adi Fledel Alon, and Gila Kamhi. Self-improving customer review response generation based on llms, 2024.

[6] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *CoRR*, abs/2006.11477, 2020.

[7] Sid Black, Gao Leo, Phil Wang, Connor Leahy, and Stella Biderman. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow, March 2021. If you use this software, please cite it using these metadata.

[8] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models, 2022.

[9] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[11] R. Lance Hogan. The historical development of program evaluation: Exploring past and present. *Online Journal for Workforce Education and Development*, 2007.

[12] Matthew Honnibal and Ines Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear, 2017.

[13] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spaCy: Industrial-strength Natural Language Processing in Python. 2020.

[14] Erik Johannes Husom, Arda Goknil, Lwin Khin Shar, and Sagar Sen. The price of prompting: Profiling energy use in large language models inference, 2024.

[15] Sujay Kapadnis. Emotion recognition dataset. `https://www.kaggle.com/datasets/sujaykapadnis/emotion-recognition-dataset?select=dataset`, 2024. Accessed: 2024-07-18.

[16] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, Wan-Teh Chang, Wei Hua, Manfred Georg, and Matthias Grundmann. Mediapipe: A framework for building perception pipelines. *CoRR*, abs/1906.08172, 2019.

[17] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[18] OpenAI. Openai api reference. `https://platform.openai.com/docs/api-reference/chat`, 2024. Accessed: 2024-07-18.

[19] Aditya Pal, Abhilash Barigidad, and Abhijit Mustafi. Imdb movie reviews dataset, 2020.

[20] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. *CoRR*, abs/1912.01703, 2019.

[21] Hasim Sak, Andrew W. Senior, and Françoise Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *INTERSPEECH*, pages 338–342, 2014.

[22] Siddharth Samsi, Dan Zhao, Joseph McDonald, Baolin Li, Adam Michaleas, Michael Jones, William Bergeron, Jeremy Kepner, Devesh Tiwari, and Vijay Gadepally. From words to watts: Benchmarking the energy costs of large language model inference, 2023.

[23] Yunlong Tang, Jing Bi, Siting Xu, Luchuan Song, Susan Liang, Teng Wang, Daoan Zhang, Jie An, Jingyang Lin, Rongyi Zhu, Ali Vosoughi, Chao Huang, Zeliang Zhang, Pinxin Liu, Mingqian Feng, Feng Zheng, Jianguo Zhang, Ping Luo, Jiebo Luo, and Chenliang Xu. Video understanding with large language models: A survey, 2024.

[24] Tapakah68. Facial emotion recognition. `https://www.kaggle.com/datasets/tapakah68/facial-emotion-recognition`, 2023. Accessed: 2024-07-18.

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[26] Raja Vavekanand and Kira Sam. Llama 3.1: An in-depth analysis of the next generation large language model. 07 2024.

[27] Mark Wickert. Real-time digital signal processing using pyaudio_helper and the ipywidgets. pages 91–98, 01 2018.

[28] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771, 2019.

[29] Weizhe Yuan, Kyunghyun Cho, and Jason Weston. System-level natural language feedback, 2024.

[30] Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola. *Dive into Deep Learning*. Cambridge University Press, 2023. `https://D2L.ai`.