A Deep Learning Platform for Language Education Research and Development

Kye Min Tan, Richeng Duan, Xin Huang, Bowei Zou, Xuan Long Do

Institute for Infocomm Research, A*STAR, Singapore

{Tan_Kye_Min, Duan_Richeng, Huang_Xin, Zou_Bowei}@i2r.a-star.edu.sg,
xuanlong001@e.ntu.edu.sg

Abstract

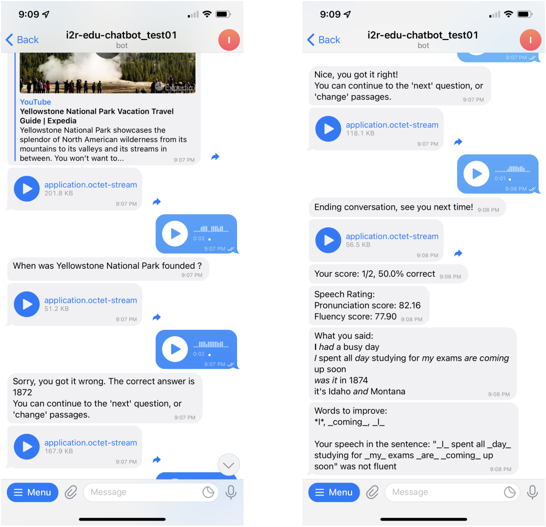
We present a language education research, development, and deployment platform that allows researchers, developers and teachers alike to easily design pipelines and collaborate on components. We also show an example use case of our platform with components such as speech evaluation, question generation, and dialog systems. To our knowledge, this is the first language education research platform which supports the integration of speech, NLP, and various custom components.

Index Terms: computer-assisted language learning (CALL), human-computer interaction, deep learning platform for language education

1. Introduction

One of the driving factors for successful language learning is the time spent immersed in that language. As it is not always possible for teachers to provide the ideal, round-the-clock learning environment for students, significant effort has been put into the research and development of tools in the language education domain. Language education is an interdisciplinary research area which involves speech technology, speech therapy, natural language processing, and educational research. Although there are many existing toolkits which serve specific domains such as Kaldi and Fairseq, there exists no such platform targeting language education. This leads to difficulties in developing an immersive language education system by adapting these fragmented tools. Significant effort is needed to conduct experiments on their individual platforms and integrate the completed components as different programming languages may be involved (Kaldi - C++ and Fairseq - Python) and libraries (PyTorch and Tensorflow).

Our experience in the field suggests that having a unique, efficient, and flexible toolkit can significantly speed up the research and development of language education techniques. With that in mind, we have designed a platform upon which an immersive language learning system can be built by consolidating all the different topics and tasks related to language education. We also developed a comprehensive set of tools with which teachers can design interactive activities, automate content design, and conduct assessments. These tools can be easily adapted to the needs of each teacher by adding the required components and adjusting the flow of data within the pipeline structure. The modular nature also gives teachers and developers the flexibility to collaborate and implement new components rapidly on a common platform, with various specialists working on different components (content design and empirical studies done by teachers, spoken language evaluation and non-native speech recognition by speech researchers, and dialog management and auto-grading tools by NLP researchers).



YouTube
Yellowstone National Park Vacation Travel Guide | Expedia
Yellowstone National Park showcases the splendor of North American wilderness from its mountains to its valleys and its streams in between. You won't want to... 9:07 PM

application.octet-stream 201.8 KB 9:07 PM

0:02 9:07 PM

When was Yellowstone National Park founded ? 9:07 PM

application.octet-stream 51.2 KB 9:07 PM

0:02 9:07 PM

Sorry, you got it wrong. The correct answer is 1872
You can continue to the 'next' question, or 'change' passages. 9:07 PM

application.octet-stream 167.9 KB 9:07 PM

9:07 PM

Nice, you got it right!
You can continue to the 'next' question, or 'change' passages. 9:07 PM

application.octet-stream 118.1 KB 9:07 PM

0:01 9:08 PM

Ending conversation, see you next time! 9:08 PM

application.octet-stream 56.5 KB 9:08 PM

Your score: 1/2, 50.0% correct 9:08 PM

Speech Rating:
Pronunciation score: 82.16
Fluency score: 77.90 9:08 PM

What you said:
I *had* a busy day
I spent all *day* studying for *my* exams *are coming* up soon
was it in 1874
it's Idaho *and* Montana 9:08 PM

Words to improve:
I, _coming_, _I_

Your speech in the sentence: "_I_ spent all _day_ studying for _my_ exams _are_ _coming_ up soon" was not fluent 9:08 PM

Figure 1: *Telegram interface showing our example system. Left is system output, right is user input.*

2. Platform

Our platform is based off the DeepPavlov framework, which is designed for the purpose of chaining components together. We adapt this framework for the language education domain by chaining speech-related components such as speech recognition and speech evaluation, NLP components such as automatic question generation and dialog systems, and other third party services. Fully trained and assembled pipelines can be deployed immediately on the included REST API server, which supports multiple users by batching incoming requests, allowing rapid prototyping and testing cycles. In addition to our support for PyTorch and Tensorflow components, we also provide a tool to convert Kaldi models into a component within our platform. In this demo, we also show the use of third party services by integrating their API calls onto the platform as individual components. This allows developers and educators to work with tools they are most familiar with.

3. Example pipeline

In this section, we will be showing off the capabilities of our platform with an example pipeline with conversational and passage comprehension aspects. As shown in Figure 1, our system chats and conducts Q&As with users through a Telegram bot that interfaces with our platform through HTTP requests. As users can communicate with the system via voice messages, students with vision impairment are also able to use our system effectively. Since all interaction takes place within the Telegram app, many users will find the chat interface familiar and easy to pick up.

3.1. Question generation

The Telegram bot first prompts (using a voice generated by TTS) the user to select an educational video to watch. After the user has finished the video, the bot will ask automatically generated questions based on the video content and assess the user's answer. To automatically extract question-worthy keywords and generate questions from a given passage, we designed an answer-unaware Factoid Question Generation (FQG) module based on two well-trained T5 models [1]. Keywords are extracted by building a semantic graph and passing its linearized output into a T5 model to predict potential answers. The proposed answers are concatenated with the passage as input to another T5 model to generate candidate questions. A BART-based machine reading comprehension model [2] is then used to filter the generated pairs by attempting to answer the questions. All FQG-related models are trained using the PyTorch toolkit.

3.2. Dialog system

The user can continue answering questions, switch to another video, or have an open-domain chat. The conversation flow is controlled by a hybrid rule-based and model-based dialog system. For the open domain chat, we use a BART-based model for its state-of-the-art performance in text generation, and fine-tune our model on the DailyDialog dataset, along with intent cues to control the conversation flow. We chose the BART-base version to keep computational requirements for running the system low.

3.3. Speech evaluation

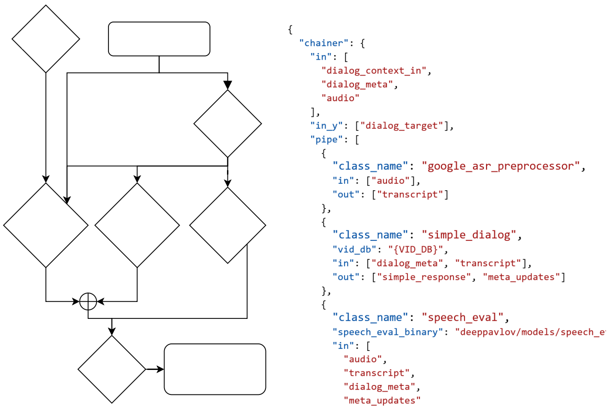
At the end of the conversation, customized feedback will be presented to the learner, including the results of the Q&A assessment and speech assessment. We utilize a lightweight version of our previously published speech evaluation engine [3]. The speech evaluation engine is based on our recently developed technique [4] for primary school students learning a second language, in which we performed unsupervised child-to-adult speech adaptation using adversarial multi-task learning. The acoustic model and the scoring model were trained with Pytorch and Kaldi respectively. Thanks to the flexibility of our platform, we are able to utilize Kaldi libraries written in C++ as part of our system. This allows us to benefit from the performance optimizations and speech-focused features of Kaldi. Finally, scoring and error statistics are compiled across the entire conversation for the final summary.

3.4. Third-party APIs

In each conversation turn, the system takes in the user's speech utterance, converts it to text, and sends it to the dialog system. Here, we use Google's speech recognition service to demonstrate the ease of integrating a third-party system on our platform. Since our pipeline system is designed with modularity in mind, third-party API calls can be converted into a native component on our platform.

4. Ongoing research and development

By building upon the flexibility of our platform, our ongoing research includes creating a multimodal dialog system together with researchers from the computer vision area, working with speech therapists to design more effective pronunciation training methods, and collaborating with school teachers to conduct empirical studies for educational evaluations.



```
{
  "chainer": {
    "in": [
      "dialog_context_in",
      "dialog_meta",
      "audio"
    ],
    "in_y": ["dialog_target"],
    "pipe": [
      {
        "class_name": "google_asr_preprocessor",
        "in": ["audio"],
        "out": ["transcript"]
      },
      {
        "class_name": "simple_dialog",
        "vid_db": "{VID_DB}",
        "in": ["dialog_meta", "transcript"],
        "out": ["simple_response", "meta_updates"]
      },
      {
        "class_name": "speech_eval",
        "speech_eval_binary": "deeppavlov/models/speech_e
        "in": [
          "audio",
          "transcript",
          "dialog_meta",
          "meta_updates"
```

(a) System diagram (b) Pipeline JSON file

Figure 2: *Our example pipeline.*

5. Conclusions

We presented a language education research, development, and deployment platform which is easy to use and is compatible with many popular programming languages and toolkits, which enables effective cross-domain R&D collaboration and translation. We showcase its abilities by providing an example conversational language learning system which allows students to practice their comprehension and conversational skills in a natural environment. Our framework allows us to easily extend functionality to incorporate various tasks while swapping out modular components as more performant ones are developed.

6. Acknowledgements

The question generation research has been supported by the NSTG project of Institute of Infocomm Research of A*STAR (CR-2021-001).

7. References

[1] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020.

[2] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 7871–7880.

[3] K. Shi, K. M. Tan, R. Duan, S. U. M. Salleh, N. F. A. Suhaimi, R. Vellu, N. T. H. H. Thai, and N. F. Chen, "Computer-Assisted Language Learning System: Automatic Speech Evaluation for Children Learning Malay and Tamil," in *Proc. Interspeech*, 2020, pp. 1019–1020.

[4] R. Duan and N. F. Chen, "Senone-aware adversarial multi-task training for unsupervised child to adult speech adaptation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 7758–7762.