Proceedings of INTERSPEECH, DOI: 10.21437/Interspeech.2019

Framework for conducting tasks requiring human assessment

Martin Grůber, Adam Chýlek, Jindřich Matoušek

New Technologies for the Information Society (NTIS)
Faculty of Applied Sciences, University of West Bohemia, Czech Republic
gruber@ntis.zcu.cz, chylek@ntis.zcu.cz, jmatouse@ntis.zcu.cz

Abstract

This paper presents a web-based framework that improves and simplifies the design and the deployment of tasks that require human input. These tasks may include speech, text or image transcription, annotation and evaluation. The focus is on listening tests for the purpose of a speech synthesis quality assessment. The framework is quite flexible, i.e. many different types of tasks can be prepared and presented to participants. The participants can then work on the tasks via a user-friendly GUI and their responses are recorded in a database. The framework is ready to be integrated as an external task for Amazon Mechanical Turk but it can also be used as a stand-alone platform.

Index Terms: listening test, speech synthesis, Mechanical Turk

1. Introduction

As speech synthesis techniques are, during their development, producing speech of many quality levels depending on what specific technique is used or what setting is applied, there is a need to compare various versions of a synthetic speech to be able to determine which technique or setting produces speech of a better quality. To get information about the synthetic speech quality, there are several types of listening tests including but not limited to MOS, MUSHRA or preference tests. To design different types of listening tests, a general tool is required.

Although the possibility to conduct various types of listening tests within a single framework was the main motivation, there are other scientific areas that require some human processing of either input or output data. Other applications of this framework might include e.g. audio and/or video transcription or annotation, text annotation, identification of objects in images, etc.

The presented web-based framework (ARTIC) allows a task designer to define a flexible structure of a task. The task consists of queries where each query can, in general, have different structure. The query is composed of samples (optional), questions (possibly related to the samples), and possible answers to these questions. The questions/answers can be of various forms like input text fields, check-boxes, radio-buttons, simple buttons with text labels, tree-like menu or numeric sliders. The design of the system emphasizes extendibility and incorporates suggestions and needs from the users and test designers of a previous more limited version of such framework.

The framework is implemented as a single page application using Angular framework for the frontend, PHP for the backend and MySQL database for storing the results, logs, and the information about participants. It's fully integrable into Amazon Mechanical Turk platform as an external task. It requires Turkers to register only once into the framework and then they are automatically logged in when accepting a task.

The task structure is described in Section 2 and the application frontend GUI in Section 3.

2. Task structure

A task is the main unit the framework works with. For example, a single listening test is a task. The task consists of queries. Each query is presented to participants on a single page with (optional) samples and questions with possible answers. Questions can be organized into groups and categories in order to create complex layouts. If a complex layout is not necessary for a query, a simplified question definition can be used. The queries can be shown to the user in a predefined order or randomized. If randomization is enabled, the order of the queries is random for each participant, i.e, each participant is presented with the queries in a different random order.

Generally, each query can have different structure, as each query is defined independently. However, for most use cases (e.g. listening tests), the structure is the same for all queries, just the samples vary. This further simplifies automatic generation of such tests (which is not a part of this framework, it's rather an extension out of this paper's scope).

2.1. Query structure

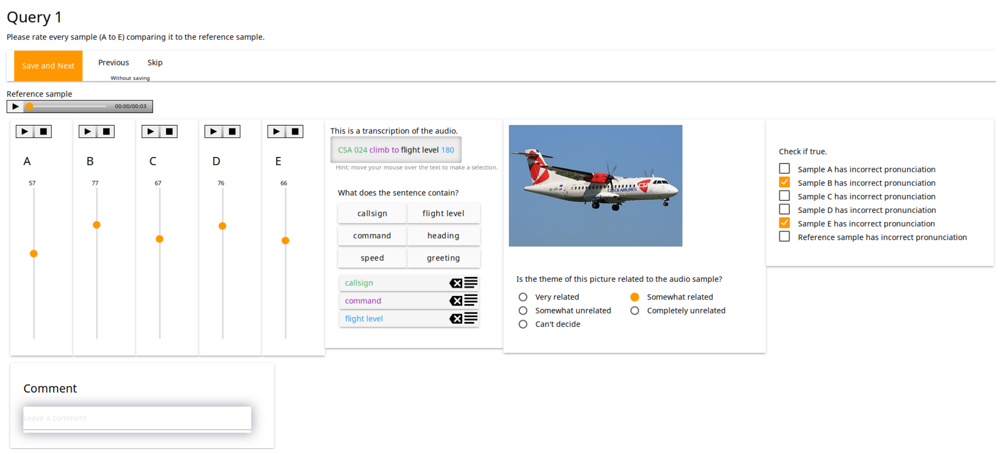
A query contains a description, samples (see Section 2.2) and questions with possible answers (see Section 2.3). For a complex layout, the query can be structured into groups. A group can also contain samples and be further structured into categories. A category then contains questions with possible answers related to the samples in the superior group. Apart from a visual separation of groups and categories, this structure allows the designer to create customized layout, choosing from horizontal, vertical or grid alignment of the contained items.

2.2. Sample types

There are several sample types a query can include. The test designer is free to use any combination of sample types and any number of samples in each query.

Mainly for the listening tests and audio transcriptions, there is an audio sample - the user is presented with a player that plays an audio file either on request or automatically. At least WAV and MP3 formats are guaranteed to work. For other formats, the availability depends on the browser.

Proceedings of INTERSPEECH, DOI: 10.21437/Interspeech.2019



Query 1

Please rate every sample (A to E) comparing it to the reference sample.

This is a transcription of the audio.

CSA 024 climb to flight level 180

Hint: move your mouse over the text to make a selection.

What does the sentence contain?

Is the theme of this picture related to the audio sample?

Very related [✓ Somewhat related] Somewhat unrelated Completely unrelated Can't decide

Check if true.

- ☐ Sample A has incorrect pronunciation
- ☑ Sample B has incorrect pronunciation
- ☐ Sample C has incorrect pronunciation
- ☐ Sample D has incorrect pronunciation
- ☑ Sample E has incorrect pronunciation
- ☐ Reference sample has incorrect pronunciation

Figure 1: Example of the ARTIC framework GUI (available also online[1]).

Targeting for example sentiment annotation, we provide a simple text sample that also supports HTML formatting. For a more complex annotation, e.g. finding named entities, there is an advanced text sample that allows the participants to select a part of the text and mark or annotate it with a label.

Concluding the sample types, we offer also a simple image. The common formats are supported, i.e. JPG, PNG, GIF, etc.

Other sample types are planned to be supported in the near future (e.g. video) or can be easily added later if necessary (e.g. selectable audio or video).

2.3. Answer types

Various answer types are available allowing the designers to choose just the right input option for their use case.

There is a text input field for an arbitrary text answer that is useful for example for transcription or comments. Restrictions on the length (minimum, maximum) can be set.

Getting an answer from a set of choices (e.g. preference tests) can be achieved using a group of labelled radio-buttons for single-choice answers or labelled check-boxes for multiple-choice answers. The minimum and the maximum number of selected check-boxes can be defined.

To get the most out of the advanced text sample type, it is possible to create a group of buttons with labels. The selected part of a sample may be marked with a label of the corresponding button. A menu or a tree-like structure of buttons is also useful. The labels of the menu are derived from a more complex, multi-level hierarchy of labels, e.g. in a 2—tier hierarchy that marks the severity of audio quality issues the participant would first chose the "pitch issue" or "stress issue" category and then has to select either "severe" or "minor" severity of the issue. The minimum and the maximum number of selections for both the buttons and the menu can be defined.

A slider can be used for numerical values, e.g. perceived quality in MUSHRA tests. The minimum and the maximum value, the increment and the orientation of the slider can be altered.

2.4. Participants

The framework can be configured to either allow participants to register themselves or use a pre-existing database of participants. A limit on the number of participants can be set. The participants can be registered either locally (within each task) or globally. Global participants can then participate in other tasks without a need to register again and they have a permanent profile storing all information they provided so far.

Each task might require the participants to meet some demands, e.g. a specific mother tongue, an age range, or a gender. If a new participant fills in information that indicates that he doesn't meet the requirements, he is not allowed to participate. If a global participant is trying to access the task, his profile is also checked whether he meets the current task requirements.

3. Frontend

The frontend is an interface used by participants to access tasks. Questions in queries can be answered using this interface. An example of a query is depicted in Figure 1 showcasing most of the available sample and answer types.

In the upper part of the page, there is a query description and buttons to submit the answers and to move through the queries. The middle part of the page is filled with samples, questions and possible answers. Several answer types are presented in this example. On the very left, there are audio samples with sliders along with a reference audio sample (MUSHRA test). On the middle left, there is an advanced text sample with a group of buttons – parts of the text sample can be selected, assigned with labels and thus highlighted (e.g. using colours). On the middle right, there is an image sample with single-choice answers (radio-buttons). There is also a panel with multiple-choice answers (check-boxes; related to the set of audio samples) on the very right. In the bottom, there is a simple text input where any arbitrary comment might be filled in.

After the participant submits the answers using the "Save and next" button, either a next query is presented or, if there isn't any next query, an end screen is shown that allows the participant to flag the answers as final. When the answers are marked as final, the user can no longer access the test and payment through the Mechanical Turk can be approved.

4. Conclusion

This paper presents an application which serves for presenting a task to participants who are supposed to go through queries and provide answers to questions defined in each query. The main purpose of this framework is to assess various types of listening tests to improve speech synthesis quality. Due to its flexibility, it can be used also in other areas requiring human assessment.

5. Acknowledgements

This research was supported by the Ministry of Education, Youth and Sports of the Czech Republic, project No. LO1506 and by the European Regional Development Fund under the project Robotics for Industry 4.0 (reg. no. CZ.02.1.01/0.0/0.0/15_003/0000470).

[1]http://artic-tests.zcu.cz/demo_interspeech_2018 → use demo/demo credentials or try to register yourself. You can watch a demo at https://youtu.be/Aq_MecqOrvk.