MLPC 2024 Task 1: Data Collection

Katharina Hoedt, Verena Praher, Paul Primus, Florian Schmid

Institute of Computational Perception
Johannes Kepler University Linz
March 4, 2024

Context

Our overall objective for this year's project is to develop a system that can detect speech commands in audio recordings. The purpose of this system is to allow users to control different devices in a smart home using speech commands. Read the *Project Description* on the Moodle page for a detailed project outline.

Developing such a keyword recognition system with machine learning entails the following steps:

1. record a training set of keywords and non-keyword sounds
2. compute a set of candidate audio features
3. perform a thorough analysis of the features and select useful features
4. train and evaluate a range of classifiers and hyperparameters; find the ones that can distinguish between keywords and unrelated sounds, such as "Other Speech/Noise" and "Silence"
5. apply and evaluate the trained classifiers for detecting speech commands in everyday scenarios and select the model that works the best

Step 1 will be a collective effort joining the forces of all individual course participants. We will assist you with step 2 and provide audio features for the recordings. Steps 3, 4, and 5 are your next tasks, which will be carried out in small teams over the remainder of the course.

Task Outline (8 + 4 bonus points)

Deadline: March 13
In this phase, we will curate datasets by collecting speech recordings. Luckily, we are a group of almost 300 students and four teachers, with likely a broad range of different recording devices (smartphones). This will help us to make our classifiers robust to different speakers and devices; as mentioned in the project description, this is one of the system's key requirements. Furthermore, there will be non-native speakers among the students, which is another positive aspect, as it will likely make our classifier more robust in correctly interpreting German speech commands spoken by non-native speakers.

Below is a step-by-step guide to complete *Task 1: Data Collection*.
Participating in Task 1 is a requirement to pass the course.

Step 0: Prerequisite - Setting up the environment for running test scripts

The following steps will assist you in setting up the environment for running the scripts to test your files before submission. You may also use this environment for converting your recording to mp3 format and to develop your keyword spotting system if you plan to use Python.

1. Install miniconda on your system by following the instructions on the official website, or these video instructions
2. Open command prompt
3. Create a conda environment using the following command:
   `conda create -n mlpc_project python=3.11`
4. Activate your environment:
   `conda activate mlpc_project`
5. ffmpeg is a library to load, modify, and convert audio between different file formats. You can simply install it via conda:
   `conda install conda-forge::ffmpeg`
6. Download the archive "mlpc_project.zip" from Moodle (item *MLPC Project*) and extract it to your favorite location.
7. Navigate to "mlpc_project" and install its requirements:
   `pip install -r requirements.txt`
8. You can test your setup by running the following command:
   `python test_split_recordings.py examples/Slideshow1.mp3 1`

The last step runs the python script `test_split_recordings.py`. The script takes as input the mp3 recording `examples/Slideshow1.mp3` and the index of the slideshow (`1` in this case). It will create a directory `mlpc_project/recordings_split/slideshow1` containing the spoken words extracted from the full recording of Slideshow 1 (see Step 4).

Step 1: Voice Recording Consent

On Moodle, you will be given three options to choose from:

- I allow recordings of my voice to be used for academic purposes.
- I allow recordings of my voice to be used only in the context of the MLPC exercise class.
- I will not provide recordings of my voice. I will do an annotation task instead.

We would highly appreciate it if you provide speech recordings, as data to learn from is the basis for every successful machine learning project. There will be NO personal information (e.g. your name) attached to the recordings, we will differentiate between speakers only by a numeric ID. If a larger group of students chooses option 3, we will not be able to curate our own dataset, and we will have to use a public pre-recorded dataset, which will be a lot less attractive. Note that if you choose option 3, you will receive an annotation task instead of a recording task, in which you will assess the quality of spoken keywords and annotate the presence of keywords in everyday scenes (in order not to be unfair to the colleagues participating in the data collection, the annotation task will require at least as much time as the collection). **If you choose option 3, you can stop after indicating your**

choice and wait for further instructions on your new task (which you will receive in about 2 weeks).

If you choose option 1, we will publish the dataset in a strictly anonymized form with an appropriate license that forbids commercial use but allows researchers around the world to develop and test machine learning algorithms on this dataset. This is a valuable contribution to the scientific community, as it allows research groups around the world to compare their machine learning methods on this new benchmark.

By choosing option 1 or 2, you also confirm that you are the only person audible in all uploaded recordings. This is a requirement, as we can not use recordings of any other person without consent.

Indicate your choice in the *Voice Recording Consent* item on Moodle.

Step 2: Familiarize yourself with the words you are about to record

Read through the list of words that will be recorded. In particular, if you are a non-native German speaker, make sure you are familiar with the pronunciation of the words. It is NOT expected that your pronunciation is perfect.

Keywords that will be recorded and the corresponding English word in parenthesis:

1. Fernseher (TV)
2. Heizung (heating)
3. Licht (lights)
4. Lüftung (ventilation)
5. Ofen (oven)
6. Alarm (alarm)
7. Radio (radio)
8. Staubsauger (vacuum cleaner)
9. an (on)
10. aus (off)

Keywords 1.-8. refer to **devices** in a smart home while 9. and 10. refer to **actions** to turn on or off a device. Therefore, a full speech command will refer to the combination of a device keyword (1. to 8.) and the desired action (9. or 10.).

Other words that will be recorded:

- Brötchen (bread roll)
- Spiegel (mirror)
- wunderbar (wonderful)
- kann (can, similar to “an”)
- Haus (house, similar to “aus”)
- nicht (not, similar to “Licht”)
- warm (warm, similar to “Alarm”)
- offen (open, similar to “Ofen”)
- Leitung (conduit, may be confused with “Heizung”)
- Schraube (screw, may be confused with “Staubsauger”)

Step 3: Prepare the recording device

Our keyword spotter is expected to perform well on audio recorded by mobile devices. Use your mobile device of choice (e.g., smartphone, smartwatch, tablet) to record your voice. In case you do not own such a device, use any other device capable of recording audio.

The final upload must be in **mp3** format. There are multiple ways to obtain an audio recording in MP3 format. Some suggestions:

- Use the `convert_to_mp3.py` script in the `mlpc_project` folder and the conda environment you have set up in Step 0. The script uses ffmpeg to convert your audio format into mp3. It can handle a lot of common audio formats.
    - Open command prompt
    - Navigate to the directory: `mlpc_project`
    - Activate conda environment: `conda activate mlpc_project`
    - Run script: `python convert_to_mp3.py <audio_file>`
    - It will create a ".mp3" file in the same folder as `<audio_file>`
- Use an app that stores your audio recording in mp3 format. There may be one pre-installed on your device. Otherwise, install one from the Android or iOS app store (e.g., search for "mp3 recorder").

Step 4: Record and upload Slideshows 1-3

On Moodle, there are three items denoted as *Slideshow 1-3* and three corresponding upload items (*Upload Slideshow 1-3*). Each slideshow contains a total of 80 words that will be recorded in a total of 4 minutes. Each word is shown for 3 seconds. Within this 3-second timeframe, you are supposed to speak the displayed word **at your normal speed of speaking**.

Within the slideshows, there are countdowns for starting and ending the recording. **Make sure to start and stop the recording on your device as accurately as possible, as splitting your spoken words relies on accurate timing! Don't pause the recording - record each slideshow in a single go. Recording yourself in a completely silent environment is not necessary; however, your voice must be the dominant sound.**

If you want to get an idea of how your recording should sound, have a look at the *Example Recording Slideshow 1* item on Moodle.

Before submitting your recordings, check if they can be split correctly:

- Open command prompt
- Navigate to the directory: `mlpc_project`
- Activate conda environment: `conda activate mlpc_project`
- Run the test script for each of the three slideshows:
  `python test_split_recordings.py <mp3_file> <slideshow_index>`
- There are three possible outcomes:
    - Successfully detected and split all words: optimal outcome
    - More than 90% of words correctly detected: quality is good enough, you can submit this recording

    - Less than 90% of words correctly detected: repeat the recording process for this slideshow
- In any case, please listen to some of the split words, which will be placed in folder `mlpc_project/recordings_split`. In particular, check whether the word in the filename matches the spoken word. If this is not the case, the temporal alignment is corrupted. Please repeat the recording process and start the recording exactly when the countdown tells you to do so.

After you have recorded and checked all three slideshows, **upload the corresponding mp3 recordings to Moodle.**

Step 5: Record and upload Everyday Scenes

By using the recorded keywords and other words from Step 4, we aim to train a classifier that distinguishes different keywords from each other, and keywords from unrelated speech or other sounds. However, we also want to test how this classifier would perform in a real scenario when it comes to detecting keywords in an everyday scene.

Of course, you are not supposed to record any private scenes of your own! Have a look at *Example Scenes 1-5* on Moodle. These are examples of how such a staged everyday scene could look like. You might want to draw inspiration from them.

Your task is to record **five different scenes**, each of which must be **between 10 and 30 seconds** long. This task requires your creativity! The content of this recording should be a natural everyday scene that might occur in a smart home. The five scenes must obey the following requirements:

- **one** of the 5 scenes contains **no speech commands**
- **four** of the 5 scenes contain **at least one speech command**
    - at least the **following** device **keywords must be included** in your recordings:
        - if the last digit of your matriculation number is in {0, 1, 2, 3, 4}: Fernseher, Heizung, Licht, Lüftung
        - if the last digit of your matriculation number is in {5, 6, 7, 8, 9}: Ofen, Alarm, Radio, Staubsauger
    - **other device keywords can be included**
    - **use full speech commands:** each of the device keywords must appear in combination with “an” or “aus” (to give the command to switch the device “on” or “off”)
    - overall, **“an” und “aus”** must appear **equally often** in the five recordings
    - speak the speech command clearly and slowly (your classifiers will thank you later)
- at least one of the 5 scenes should contain speech other than the commands, you can use any language you like for the additional speech (English, Austrian dialects, …)
- **important**: make sure that no voices other than yourself can be heard in the recordings; otherwise, we would need their consent to use the recording

In addition to the five scenes (five mp3 files), you will submit a “description.txt” file. This file contains one line for each submitted scene containing:

- the filename of the recorded scene: <filename>.mp3
- a flag speech=[“True”, “False”], indicating whether speech other than the spoken commands is present in the recording
- the speech commands occurring in your recorded scenes in the correct order

A line in your “description.txt” file has exactly the following format:

<filename>.mp3: speech=[“True”, “False”]; (<device_keyword> ["an", "aus"],)*

Example: A student has the matriculation number 12345678 and will therefore use the device keywords: Ofen, Alarm, Radio, and Staubsauger. The student produces the following five recordings with the respective speech commands:

- Scene1.mp3: contains no speech other than the command: Ofen an
- Scene2.mp3: contains speech but no commands
- Scene3.mp3: contains no speech other than two commands: Radio aus, Licht aus
- Scene4.mp3: contains speech and the two commands: Staubsauger an, Alarm an
- Scene5.mp3: contains speech and the command: Heizung aus

The student verifies that the recordings fulfill all requirements defined above and writes the “description.txt” file with the following content:

```
Scene1.mp3: speech=False; Ofen an
Scene2.mp3: speech=True;
Scene3.mp3: speech=False; Radio aus, Licht aus
Scene4.mp3: speech=True; Staubsauger an, Alarm an
Scene5.mp3: speech=True; Heizung aus
```

Example Scenes 1-5 on Moodle are example recordings of the five scenes described above.

Before uploading the files, verify that your submission is correct:

- Collect the five mp3 files and the `description.txt` file in one folder
- Navigate to the directory: `mlpc_project`
- Activate conda environment: `conda activate mlpc_project`
- Run the test script specifying the submission folder as an argument: `python test_scene_submission.py <submission_folder>`
- The script will indicate possible problems with your mp3 files or the format of your `description.txt` file. Fix the problems before submitting your files to Moodle.

Finally, all five recordings and the “description.txt” file must be uploaded to Moodle, item *Upload Everyday Scenes*.

Step 6 (optional, bonus points): More Everyday Scenes

Repeat Step 5 and produce 5 more recordings of everyday scenes. In total, you will record 10 scenes that (1) use all eight device keywords at least once, (2) include two recordings

that contain no speech commands, and (3) include at least two recordings that contain speech other than the speech commands.

You will submit a total of 11 files: 10 mp3 files and the “description.txt” file containing 10 lines.

Summary

- **Completing Task 1 (Step 0 - Step 5) is a requirement to pass this course.**
- Indicate your choice in the *Voice Recording Consent* item on Moodle. If you choose option 3 and do not provide recordings, you are done for now, and you will receive a different task at a later point in time.
- Record yourself speaking the words displayed on the Slideshows 1-3. Pay special attention to pressing the record button on your device when the countdowns in the slideshows tell you to do so. You should end up with three recordings that are almost exactly 4 minutes and 7 seconds in length.
- Upload the three mp3 recordings for *Slideshows 1-3* to the respective items on Moodle (*Upload Slideshows 1-3*).
- Record 5 everyday scenarios. The exact specifications are outlined in Step 5.
- Create a “description.txt” file describing your recorded scenes' content. The exact format is outlined in Step 5.
- Upload the five everyday scenes (ten if you want to get the bonus points) in mp3 format, and the “description.txt” file to the item *Upload Everyday Scenes* on Moodle.
- Use the provided scripts `test_split_recordings.py` and `test_scene_submission.py` to verify that your slideshow recordings can be correctly split and that your scene submission meets all specifications, before submitting the files.
- If your recordings pass the test scripts and the spoken words are correctly aligned with the filenames, **you will receive full points on this task.**