67355 Introduction to Speech and Audio Processing

Final Project

For the final project in the *Introduction to Speech and Audio Processing*, course, no. 67355, you will experience building a full Automatic Speech Recognition (ASR) pipeline.

The exercise should be done **in pairs** or in **triplets**, and is to be submitted via moodle by the deadline appearing under the submission box.

See submission guidelines for further instructions.

1 Final Project

Your task in the final project is to build the best ASR system. You can use everything we have learned in the course including:

1. Dynamic Time Warping (DTW)
2. Hidden Markov Models (HMMs)
3. DNN-HMM
4. End-to-End models: CTC, ASG, RNN Transducers, LAS, etc.
5. Language modeling
6. Different acoustic feature
7. Ensemble methods
8. Different search methods
9. etc.

Although your goal is to build the best ASR system you can, you should experiment and gain experience with as many of the technologies we went over in the course. In case you do not have access to a GPU, you should use Google collab. You can not use pre-trained ASR models (e.g., Whisper).

You should document evaluate, analyze, and document the performance of each of the tested configurations, that way you will make smarter decisions when picking the final model. Your model can be a single model, an ensemble of models, etc.

You can consider this as a full-blown research project (on a small scale). In which we ask, what component/configuration has the most impact on ASR performance.

2 Dataset

To build the ASR model you should use the AN4 dataset. This is a small dataset recorded and distributed by Carnegie Mellon University (CMU). It consists of recordings of people spelling out addresses, names, etc.

You should download the dataset using the following link `https://drive.google.com/file/d/1MiPqJDm6gX_ayXZJ2LHeUbG0UNZfNagF/view?usp=sharing`. After downloading you will find a split of the dataset into `train / test / val` splits.

Notice: the vocabulary of the dataset is not super big. You are allowed to leverage this information if you believe it will improve the model's performance.

3 Evaluation

You should evaluate your model using Word Error Rate (WER) and Character Error Rate (CER). To keep eveluation across the different teams consistent, you should use the following package only: JiWER `https://pypi.org/project/jiwer/`.

```python
from jiwer import wer

reference = "hello world"
hypothesis = "hello duck"

error = wer(reference, hypothesis)
```

4 What should we submit?

You should submit your source code, trained model (with the ability to run it per request), and a **detailed report**. Your report should include every configuration your experimented with: i.e., what was working and what was not, results for the different configurations you tried, model performance, graphs, table of results, etc.

In other words, the report should make it clear what you tried, what was the performance for each evaluated setup, and what led you to choose this specific model. Results should be reported for both train, val, and test splits.

Grading will be based on (i) model performance; and more importantly (ii) the report! I encourage you to explore different models, configurations, etc.

Although you submit your work in pairs/triplets, grading will be per individual. Each team will have 20 min in-person meeting to present what they did. Each team will be asked to defend their model, explain what they did, what tools they used, and what was working and what did not work.

5 Submission Guidelines

- You should submit your source code.
- Pre-trained models.
- Report including your names and IDs.
- Please submit a single zip/tar file containing all relevant files.

Notice: we will not train your models nor run inference on our side. We will mainly look at the code, and read the report. However, we might ask you to run inference during the 20min project presentation, so be prepared for that.