10 Academy Batch 4- Weekly Challenge: Week 4

African language Speech Recognition - Speech-to-Text

Overview

Business Need

Speech recognition technology allows for hands-free control of smartphones, speakers, and even vehicles in a wide variety of languages. Companies have moved towards the goal of enabling machines to understand and respond to more and more of our verbalized commands. There are many matured speech recognition systems available, such as Google Assistant, Amazon Alexa, and Apple's Siri. However, all of those voice assistants work for limited languages only.

The World Food Program wants to deploy an intelligent form that collects nutritional information of food bought and sold at markets in two different countries in Africa - Ethiopia and Kenya. The design of this intelligent form requires selected people to install an app on their mobile phone, and whenever they buy food, they use their voice to activate the app to register the list of items they just bought in their own language. The intelligent systems in the app are expected to live to transcribe the speech-to-text and organize the information in an easy-to-process way in a database.

You work for the Tenacious data science consultancy, which is chosen to deliver speech-to-text technology for two languages: Amharic and Swahili. Your responsibility is to build a deep learning model that is capable of transcribing a speech to text. The model you produce should be accurate and is robust against background noise.

Data

- Dataset for Amharic here
- Dataset for Swahili here

Data Features

- Input features (X): audio clips of spoken words
- Target labels (y): a text transcript of what was spoken

Expected Outcomes

Skills:

- Working with audio as well as text files
- Familiarity with the deep learning architecture
- Model management (building ML catalog containing models, feature labels, and training model version)
- MLOps with DVC, CML, and MLFlow

Knowledge:

- Audio and text processing
- Deep learning methods (TensorFlow, Keras, Pytorch)
- Hyperparameter tuning
- Model comparison & selection
- Experiment Analysis

Communication:

- Reporting - writing with a standardized format

Competency Mapping

The tasks you will carry out in this week's challenge will contribute differently to the 17 competencies 10 Academy identified as essential for job preparedness in the field of data science, and Machine Learning engineering. The mapping below shows the change (lift) one can obtain through delivering the highest performance in these tasks.

MC0: Marginal contribution - causes no significant change

MC1: Minor contribution - recognized for routine performance gain

MC2: Measurable contribution - will contribute a value towards portfolio and job readiness metric

MC3: Major contribution - the best performance of these types of tasks at least three times within our training leads one to attain a job-ready level along that competency dimension.

Competency	Value	Potential contributions from this week
Business Understanding	MC2	Understanding and reasoning the business context. Thinking about suitable analysis that matches the business need. Thinking about clients and their interests.
Data Engineering	MC3	Thinking about how to store data for easy analysis, and what format to use to build responsive dashboards.
Data Understanding	MC3	Understanding the data provided and extract insight. Exploring different techniques, algorithms, statistical distributions, sampling, and visualization techniques to gain insight.
Dashboard & Visualization	MC2	Building a dashboard to explore data as well as to communicate insight. Advanced use of modules such as plotly, seaborn, matplotlib etc. to build descriptive visualizations. Reading through the modules documentation to expand your skill set.
Mathematics and Statistics	MC2	Thinking about statistical distributions, sampling, bias, overfitting, correlations.

MLOps & Continuous Delivery	MC3	Using Github for code development, thinking about feature store, planning analysis pipeline, using MLOps tools for code, data, model, artifact versioning, setting up docker containers for automated microservice deployment.
Modeling and evaluation	MC3	Comparing multiple Deep learning techniques; training and validating DL models; choosing appropriate architecture, loss function, and regularisers; hyperparameter tuning; choosing suitable evaluation metrics.
Python programming	MC3	Advanced object-oriented python programming. Python package building.
SQL programming	MC1	Building feature stores using SQL or NoSQL databases.
Fluency in the Scientific Method	MC2	Thinking about evidence. Generating hypothesis, testing hypothesis. Thinking about different types of errors.
Ethics	MC3	data privacy, data security, ethical use of data. The 8 principles of responsible machine learning
Statistical & Critical Thinking	MC2	Thinking about the difference between causal vs chance correlation. Giving reasonable recommendations. Thinking about uncertainties.
Software Engineering & Dev Environment	MC2	Reading articles on software project planning. Unit testing.
Impact & Lifelong learning	MC3	Learning new concepts, ideas, and skills fast, and applying them to the problem at hand.
Professional Culture & Communication	MC2	Writing a well-formatted presentation with no mistakes, formatted nicely.
Social Intelligence & Mentorship	MC2	Asking for help early, providing help for those who need it, avoiding being stuck.
Career Thinking	MC3	Working within groups in a successful way

Team

Instructors: Yabebal, Abubakar, Mahlet, Kevin

Key Dates

- **Discussion on the case** - 11:30 UTC time on Monday 02 August 2021. Use #all-week4 to ask questions.
- **Interim Submission 1** - 8:00PM UTC time on Wednesday 04 August 2021.
- **Interim Submission 2** -8:00PM UTC time on Saturday 07 August 2021
- **Final Submission** - 8:00PM UTC time on Thursday 12 August 2021

Leaderboard for the week

There are 100 points available for the week.

20 points - community growth and peer support.

13 points - technical public and group-based RC channels

- Total number of messages (5)
- Total number of Mentions (3)
- Total number of DM connections (5)

7 points - community activities

- Number of messages in non-technical channels (4)
- On-time presence in Gmeet sessions (3)

30 points - presentation and reporting.

15 points - interim submission. This is measured through:

- Survey of speech-to-text deep learning architectures (4)
- Discussion of data formatting and pre-processing (e.g. mle-spectrum, character representation, etc.) (4)
- Discussion of the loss function, evaluation metrics, and other key components (4)
- Discussion of Ethics, bias, and fairness (3)

15 points for the final submission. This is measured through:

- PDF of a Slide/Report/Blog Entry submission is a must.
- Evidence to publication or submission of the report in a blog e.g. medium, LinkedIn, or other similar platforms (4 points)
- Clarity of message, writing structure e.g. logically separated sections, and appropriate usage of graphs (4 points)
- Professionalism/production value (free of spelling errors, use of same font, well-produced) (2 points)
- Discussion of Ethics, bias, and fairness and attempts tried to address them (5)

50 points - Technical content

20 points - Interim submission

1. Github link submission (20)
    - Object-oriented programming (3)
    - Well written Readme (3)
    - Jupyter notebook that shows the visualization of the data (3)
    - The first implementation of the DL architecture (3)
    - Git branches demonstrating team working on different components simultaneously (4)
    - Pull Request history with CML modified messages (4)

30 points - Final submission

- MLOps setup (10)
    - Working CML git workflow implementation in the repo (5)
    - Screenshot of example CML report when a git push a code (5)
- Github Link submission (10)
    - Deep Learning Model implementation (10)
- Dashboard (10)
    - Screenshot of Dashboard (6)
    - Deployed model or video of the deployed model in action (4)

Badges

Each week, one user will be awarded one of the badges below for the best performance in the category below.

In addition to being the badge holder for that badge, each badge winner will get +20 points to the overall score.

Visualization - the quality of visualizations, understandability, skimmability, choice of visualization
Quality of code - reliability, maintainability, efficiency, commenting - in the future this will be CICD/CML
An innovative approach to analysis -using latest algorithms, adding in research paper content and other innovative approaches
Writing and presentation - clarity of written outputs, clarity of slides, overall production value
Most supportive in the community - helping others, adding links, tutoring those struggling

The goal of this approach is to support and reward expertise in different parts of the Machine learning engineering toolbox.

Group Work Policy

Everyone has to submit all their work individually.

This week, however, you are expected to complete the project with your assigned group. In the table below, your name is assigned to one of the groups we formed. You are expected to collaborate very closely with your team and finish all tasks.
The groups are divided into 6 and 9 per group

- **Group 1, 4 and 5** will work with the **Swahili** dataset
- **Group 2, 3 and 6** will work on the **Amharic** dataset

Group	Name

1	Steshy Kibika Solo Ache Germain Rukundo Blaise Papa Amon Kimutai Desmond Onam David Mukuzi Boris Papineau Hirwa jakinda oluoch
2	Milky Bekele Dibora Gebreyohannes Eyerusalem Gebreegzabiher Yosef Engdawork Saba Agizew Woldeamanuel smegnsh demelash Abel Mitiku Chimdessa Tesfaye Hordofa Haftom Tekleweyni
3	Michael Tekle Behigu Gizachew Fumbani Banda Natnael Sisay Robert Ssebudandi christian ZANOU Elias Andualem Ethani Caphace
4	Khairat Ayinde Toyin Hawau Olamide Mubarak Sani Muhammed Kutashi Jocelyne Mukamisha Same Michael Michael Darko Ahwireng sibitenda harriet Nebiyu Samuel smegnsh demelash
5	Zelalem Getahun Abreham Gessesse Luel Hagos Kate Njoki Binyam Sisay mizan abaynew BEZAWITALEM YIMER Maelaf Estiphanos

	Euel Fantaye Bereket Kibru
6	Daniel Zelalem Yosef Alemneh Azaria Tamrat Dorothy Cheruiyot Bethelhem Sisay UWASE Rachel Elizabeth Nanjala Eliphaz Niyodusenga Natneal Teshome

Within your group, you can share concepts, references, codes, figures, and have similar answers to the questions. You **MUST** write both the interim and final reports yourself, submit Github links from your Github account, and take screenshots from your computer. All group members may have similar code in their git repository, but you should have at least run the code in your local machine and make frequent commits. Making git branches, doing Pull Requests to merge branches, and following other good software development practices is mandatory.

We expect all group members to contribute equally. We leave the assignment of roles within groups to the group members.

Late Submission Policy

Our goal is to prepare successful learners for the work and submitting late when given enough notice, shouldn't be necessary.

For interim submissions, those submitted 1-6 hours late will receive a maximum of 50% of the total possible grade. Those submitted >6 hours late may receive feedback, but will not receive a grade.

For final submissions, those submitted 1-24 hours late, will receive a maximum of 50% of the total possible grade. Those submitted >24 hours late may receive feedback, but will not receive a grade.
When calculating the leaderboard score:

- This week one of your lowest week's scores will not be considered.
- From week 8 onwards, your two lowest weeks' scores will not be considered.

Instructions

The workflow for this week's challenge is as follows

- Read instructions and understand the business needs, the type of data available, the data science processed that needs to be carried out, the MLOps requirements, and the resources required/available to complete the project
- Plan the work as a team - divide the work and assign roles and responsibilities to different team members
- Setup team git repository, DVC, MLFlow, and CML integration
- Explore data, visualize, transform, and apply data augmentation techniques
- Review the pros and cons of different tools - Keras, PyTorch, Tensorflow
- Choose Deep Learning architecture, loss function, evaluation metrics, and other necessary variables.
- Implement architecture, or if using existing implementation understand the package in detail and adapt it to your needs as necessary.
- Train, test and evaluate. Study the effect of hyperparameters.
- Discuss issues and challenges including: Is your model fair? Does it have biases? What can be done to improve these ethical issues?
- Deploy model through an easy-to-use dashboard. Consider integrating MLMonitor for monitoring model performance and data drift.

Task 1 - Data pre-processing

In this part of the project process, you are required to load, understand the types and formats, explore, and visualize the data. Make sure you have a clear understanding of the inputs and outputs of your model. You may need to do data augmentation to train a model that is robust against small changes.

Here are some tasks you should do as a minimum requirement

- Load audio file
- Load transcriptions
- Convert into channels
    - Some of the sound files are mono (ie. 1 audio channel) while most of them are stereo (ie. 2 audio channels). Since the Neural network model expects all items to have the same dimensions, we will convert the mono files to stereo, by duplicating the first channel to the second

- Standardize sampling rate
    - We must standardize and convert all audio to the same sampling rate so that all arrays have the same dimensions.
- Resize to the same length
    - Resize to get an equal-sized audio sample by extending duration by padding it with silence, or by truncating it.
- Data argumentation
    - Perform data augmentation on the raw audio signal by applying a Time Shift to shift the audio to the left or the right by a random amount.
- Feature extraction: Speech recognition methods derive features from the audio, such as Spectrogram or Mel Frequency Cepstrum (MFCC).
    - Convert the augmented audio to a Mel Spectrogram.
- Acoustic modeling: After features are extracted, these vectors are passed to acoustic models. An acoustic model attempts to map the audio signal to the basic units of speech such as phonemes or graphemes.

Task 2 - Modelling and Deployment using MLOps

- Modeling: Build a Deep learning model that converts speech to text.
- Choose one of deep learning architecture for speech recognition
    - Use Connectionist Temporal Classification Algorithm for training and inference
    - CTC takes the character probabilities output of the last hidden layer and derives the correct sequence of characters
- Evaluate your model.
- Effect of data augmentation: apply different data augmentation techniques and version all of them in DVC. Train model for using these data and study the effect of data augmentation on the generalization of the model.
- Model space exploration: using hyperparameter optimization and by slightly modifying the architecture e.g. increasing and decreasing the number of layers to find the best model.
- Write test units that can be run with CML that will help code reviewers accept Pull Requests (PRs) based on performance gain and other crucial elements.
- Version different models and track performance through MLFlow

- Evaluate the model using evaluation metrics for speech recognition Word error rate (WER)
    - WER compares the predicted output and the target transcript, word by word (or character by character) to figure out the number of differences between them.

Task 3 - Serving predictions on a web interface

Use one of the platforms of your choice (Flask, Streamlit, pure javascript, etc.) to design, and build a backend to make inference using your trained model and input parameters collected through a frontend interface.

Your dashboard should provide an easy way for a user (in this case managers of the stores) to enter required input parameters, and output the predicted sales amount and customer numbers.

- Package your model as a docker container for easy deployment in the cloud.

Task 4 - Interpretation & Reporting

Report all your finding your report should include:

- Why have you chosen a specific method to process the raw dataset?
- Why have you chosen a specific method to extract features?
- Discuss each architecture you have used to build the Deep learning model.
- Comparison report on the performance of each model

N.B for reporting

Your report should start with the Introduction, the overall body of your report, and conclusion.

Submission

Interim 1: Due Wednesday 04.08 8pm UTC

1. PDF report max 2 pages. Your report should include
    a. Survey of speech-to-text deep learning architectures

    b. Discussion of data formatting and pre-processing (e.g. mle-spectrum, character representation, etc.)
2. Github link submission that demonstrates
    a. Object-oriented programming
    b. Well written Readme
    c. Jupyter notebook that shows the visualization of the data
    d. Git branches demonstrating team working on different components simultaneously

Interim 2: Due Saturday 07.08 8pm UTC

3. PDF report max 2 pages. Your report should include
    a. Discussion of the loss function, evaluation metrics, and other key components
    b. Discussion of Ethics, bias, and fairness
4. Github link submission that demonstrates
    a. The first implementation of the DL architecture
    b. Git branches demonstrating team working on different components simultaneously
    c. Pull Request history with CML modified messages
5. Screenshots
    a. Screenshots showing CML driven Pull Request and integration
    b. Screenshots showing DVC and MLFlow setup

Feedback

You may not receive detailed comments on your interim submission, but will receive a grade.

Final Due Thursday 12.08 8pm UTC

1. PDF of a Slide/Report/Blog Entry 4-6 pages report. Your report should include
    a. Abstract - summary of what has been done, what results got obtained, what is expected in the report/blog.
    b. Motivation - why the work done is important. Relating to the business to a general context.

    c. Review of previous works - what other people have done in the same or related data. What is the state of the art speech-to-text implementation for the language you are working on; the similarities and differences of your work with state of the art implementations.
    d. Data and preprocessing techniques used in the current analysis
    e. The deep learning architecture used and discussion on why that architecture is chosen.
    f. MLOps setup
    g. Modelling results based on all explorations done.
    h. Section on bias, fairness, and what is done to mitigate issues discovered.
    i. Future work - what can be improved with more time and resources?
2. Link to your Github code that includes your Jupyter notebook.
    a. Deep Learning Model implementation
    b. Working CML git workflow implementation in the repo
3. Link to your Deployed Application and/or screenshot of your dashboard
    a. Screenshots demonstrating anything you have done but that are not included in the report or github repository.
    b. Screenshot of example CML report when a git push a code
    c. Screenshot of Dashboard
    d. If the model is not deployed in a public space, attach a video of a screen record which shows the model in action.

Feedback

You will receive comments/feedback in addition to a grade.

Reference

Papers and Blogs

- OkwuGbé: End-to-End Speech Recognition for Fon and Igbo 2103.07762.pdf (arxiv.org)
- Basics of Signal Processing
- Commonly used Acoustic modeling techniques
- Introduction to Librosa
- Feature extraction with librosa

- Music feature extraction
- 3 Deep learning framework for speech recognition
- Fourier Transform with Scipy
- Week 4 folder reading materials

Git Packages

- bonaventuredossou/fonasr: End2End Automatic Speech Recognition Model in PyTorch for Fon (github.com)
- tilayealemu/MelaNet: Amharic speech recognition using Deep Learning (github.com)

Creating ssh keys

- https://www.digitalocean.com/community/tutorials/how-to-set-up-ssh-keys-2
- https://phoenixnap.com/kb/generate-ssh-key-windows-10