joe-lin-tech / emg2qwerty Public
forked from facebookresearch/emg2qwerty

Notifications Fork 7 Star 3

Code Pull requests Actions Projects Security Insights

main Go to file Code

This branch is 5 commits ahead of facebookresearch/emg2qwerty:main.

joe-lin-tech Merge branch 'main' of github.com/joe-li... 2aa3434 · 2 weeks ago 11 Commits

.github/workflows	Initial commit	5 months ago
config	init: update config files for data subset	3 weeks ago
emg2qwerty	Initial commit	5 months ago
models	Initial commit	5 months ago
scripts	Initial commit	5 months ago
.gitattributes	Initial commit	5 months ago
.gitignore	Initial commit	5 months ago
.pre-commit-config.yaml	Initial commit	5 months ago
CODE_OF_CONDUCT.md	Initial commit	5 months ago
CONTRIBUTING.md	Initial commit	5 months ago
Colab_setup.ipynb	Add files via upload	2 weeks ago
LICENSE	Initial commit	5 months ago
README.md	update: modify README	2 weeks ago
environment.yml	Initial commit	5 months ago
requirements.txt	Add files via upload	2 weeks ago
setup.cfg	Initial commit	5 months ago
setup.py	Initial commit	5 months ago

README Code of conduct License

C147/247 Final Project

Winter 2025 - *Professor Jonathan Kao*

This course project is built upon the emg2qwerty work from Meta. The first section of this README provides some guidance for working with the repo and contains a running list of FAQs. **Note that the rest of the README is from the original repo and we encourage you to take a look at their work.**

Guiding Tips + FAQs

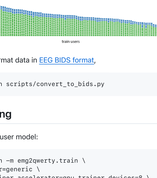
Last updated 2/13/2025

- Read through the Project Guidelines to ensure that you have a clear understanding of what we expect
- Familiarize yourself with the prediction task and get a high-level understanding of their base architecture (it would be beneficial to read about CTC loss)
- Get comfortable with the codebase
  - `lightning.py` + `modules.py` - where most of your model architecture development will take place
  - `data.py` - defines PyTorch dataset (likely will not need to touch this much)
  - `transforms.py` - implement more data transforms and other preprocessing techniques
  - `config/*.yaml` - modify model hyperparameters and PyTorch Lightning training configuration
    - **Q: How do we update these configuration files?** A: Note the structure of YAML files include basic key-value pairs (i.e. `<key>: <value>`) and hierarchical structure. So, for instance, if we wanted to update the `mlp_features` hyperparameter of the `TDSConvCTCModule`, we would change the value at line 5 of `config/model/tds_conv_ctc.yaml` (under `module`). *Read more details here.*
    - **Q: Where do we configure data splitting?** A: Refer to `config/user/single_user.yaml`. Be careful with your edits, so that you don't accidentally move the test data into your training set.

emg2qwerty

[Paper] [Dataset] [Blog] [BibTeX]

A dataset of surface electromyography (sEMG) recordings while touch typing on a QWERTY keyboard with ground-truth, benchmarks and baselines.



Setup

```
# Install [git-lfs](https://git-lfs.github.com/) (for pretrained
git lfs install

# Clone the repo, setup environment, and install local package
git clone git@github.com:facebookresearch/emg2qwerty.git ~/emg2qw
cd ~/emg2qwerty
conda env create -f environment.yml
conda activate emg2qwerty
pip install -e .

# Download the dataset, extract, and symlink to ~/emg2qwerty/data
cd ~ && wget https://fb-ctrl-oss.s3.amazonaws.com/emg2qwerty/emg2
tar -xvzf emg2qwerty-data-2021-08.tar.gz
ln -s ~/emg2qwerty-data-2021-08 ~/emg2qwerty/data
```

Data

The dataset consists of 1,136 files in total - 1,135 session files spanning 108 users and 346 hours of recording, and one `metadata.csv` file. Each session file is in a simple HDF5 format and includes the left and right sEMG signal data, prompted text, keylogger ground-truth, and their corresponding timestamps.
`emg2qwerty.data.EMGSessionData` offers a programmatic read-only interface into the HDF5 session files.

To load the `metadata.csv` file and print dataset statistics,

```
python scripts/print_dataset_stats.py
```

Total subjects	108
Total sessions	1,135
Avg sessions per subject	10
Max sessions per subject	18
Min sessions per subject	1
Total duration	346.4 hours
Avg duration per subject	3.2 hours
Max duration per subject	6.3 hours
Min duration per subject	15.3 minutes
Avg duration per session	18.0 minutes
Max duration per session	47.5 minutes
Min duration per session	9.5 minutes
Avg typing rate per subject	265 keys/min
Max typing rate per subject	439 keys/min
Min typing rate per subject	130 keys/min
Total keystrokes	5,262,671

To re-generate data splits,

```
python scripts/generate_splits.py
```

The following figure visualizes the dataset splits for training, validation and testing of generic and personalized user models. Refer to the paper for details of the benchmark setup and data splits.

To re-format data in EEG BIDS format,

```
python scripts/convert_to_bids.py
```

Training

Generic user model:

```
python -m emg2qwerty.train \
  user=generic \
  trainer.accelerator=gpu trainer.devices=8 \
  --multirun
```

Personalized user models:

```
python -m emg2qwerty.train \
  user="single_user" \
  trainer.accelerator=gpu trainer.devices=1
```

If you are using a Slurm cluster, include "cluster=slurm" override in the argument list of above commands to pick up `config/cluster/slurm.yaml`. This overrides the Hydra Launcher to use Submitit plugin. Refer to Hydra documentation for the list of available launcher plugins if you are not using a Slurm cluster.

Testing

Greedy decoding:

```
python -m emg2qwerty.train \
  user="glob(user*)" \
  checkpoint="${HOME}/emg2qwerty/models/personalized-finetuned/\$
  train=False trainer.accelerator=cpu \
  decoder=ctc_greedy \
  hydra.launcher.mem_gb=64 \
  --multirun
```

Beam-search decoding with 6-gram character-level language model:

```
python -m emg2qwerty.train \
  user="glob(user*)" \
  checkpoint="${HOME}/emg2qwerty/models/personalized-finetuned/\$
  train=False trainer.accelerator=cpu \
  decoder=ctc_beam \
  hydra.launcher.mem_gb=64 \
  --multirun
```

The 6-gram character-level language model, used by the first-pass beam-search decoder above, is generated from WikiText-103 raw dataset, and built using KenLM. The LM is available under `models/lm/`, both in the binary format, and the human-readable ARPA format. These can be regenerated as follows:

1. Build kenlm from source: https://github.com/kpu/kenlm#compiling
2. Run `./scripts/lm/build_char_lm.sh <ngram_order>`

License

emg2qwerty is CC-BY-NC-4.0 licensed, as found in the LICENSE file.

Citing emg2qwerty

```
@misc{sivakumar2024emg2qwertylargedatasetbaselines,
      title={emg2qwerty: A Large Dataset with Baselines for Touch
      author={Viswanath Sivakumar and Jeffrey Seely and Alan Du a
      year={2024},
      eprint={2410.20081},
      archivePrefix={arXiv},
      primaryClass={cs.LG},
      url={https://arxiv.org/abs/2410.20081},
}
```

About

A surface electromyographic (sEMG) touch typing dataset with baselines. https://arxiv.org/abs/2410.20081.

Readme
View license
Code of conduct
Activity
3 stars
0 watching
7 forks
Report repository

Releases

No releases published

Packages

No packages published

Languages

Python 96.8% Jupyter Notebook 1.8% Shell 1.4%

data py connects to data _ IDF