Final Project - Connecting Language to fMRI reads, Stat 215A, Fall 2025

These are the **final project specific** instructions. Please also see the general lab instructions in **lab-instructions.pdf**.

Contents

1 Submission

Push a folder called `final-project` to your `stat-215-a GitHub` repository by **23:59 on Friday December 12**. Unlike lab4, **only one group member needs to submit the project**. I will run a script that will pull from all of your GitHub repositories promptly at midnight so take care not to be late as late labs will not be accepted.

The page limit for this lab is 20 pages. The bibliography and academic honesty sections don't count towards this limit.

Follow the general lab instructions in stat-215-a-gsi/disc/week1/lab-instructions.pdf for more details. Please do not try to make your project fit the requirements at the last minute!

I have not provided a template as a guideline for the writeup. You should have clean code that would be clear to reproduce and a report structure that is compelling to read and roughly follows the conceptual progression of previous labs.

Special coding instructions

For any ridge regression model you train, please save the trained model in your results directory. It should be a .pkl file.

2 Academic honesty and teamwork

Academic honesty statement

You can use your statement from lab1 or lab2 or lab3 or lab4

Collaboration policy

Within-group: In a file named `collaboration.txt` in your `report` directory, you must include a statement detailing the contributions of each student, which indicates who worked on what. After the labs are submitted, I will also ask you to privately review your group members' contributions.

With other people: You are welcome to discuss ideas with me. Please avoid discussing specific ideas with other groups, as I don't want to receive 5 of the same lab. If you do consult with students in other groups, you must acknowledge these students in your lab report.

LLM usage policy

You are allowed to use LLMs (ChatGPT, GitHub Copilot, etc.) in accordance with the general LLM usage policy for this class. Please include a section at the end of your report briefly outlining at what stages you used LMMs and for what purpose.

3 Final Project Overview

Goal The final project focuses on predicting blood-oxygen level across brain voxels 1 via an FMRI as subjects listen to various podcasts. Specifically, the goal is to create embeddings from the text of the podcasts to predict voxels. Overall your tasks will include:

- Using NLP techniques to extract text embeddings including using/fine-tuning pre-trained large scale LLMs.
- Training a linear model to predict voxels
- Interpreting results via different techniques

Scientific Motivation A key aspect of intelligence is the ability of our brains to process rich, complicated language. Accurately predicting how the brain responds to textual stimuli implies that we have a model that could be used to understand how the brain processes language. Further, embeddings from this model can be used to understand representations in the brain. For example, these embeddings can be used to derive insights about how different parts of the brain function to process language. As a result, scientists have dedicated significant effort to measuring brain function via FMRIs. This lab focuses on experiments conduced by Alexander Huth and his lab at UT Austin.

Data

We have data from two subjects listening to stories measured by the Huth Lab at UT Austin [1].The data for each subject consists of whole-brain blood-oxygen dependent (BOLD) signals measured at various points across the podcast. That is, for each subject-story pair, we have measurements $Y \in \mathbf{R}(T' \times V)$, where T' represents the number of FMRI measurements, and V is the number of voxels. For each subject-story pair, their data can be found on the dropbox link below. Note that these files are really big, $\approx$ 20GB each. We also have the raw text for each story on the link above which we will use to generate our embeddings to predict the matrix Y .

Access data at the following link: `https://www.dropbox.com/scl/fo/6taux72r36h9wgv0mza8k/AC04j1WSCT96F`
`rlkey=4wpvdb61n3ppl307q77czwl7j&st=j1dyyoic&dl=0`

Coding

We provide code to help you process some of the FMRI data. You will find a ridge utils folder and under code/preprocessing.py, you will find multiple functions that you will use below to process the data.

You will also find a very simple code skeleton for BERT Fine-tuning.

4 Instructions

Find Word Embeddings

For this project you will have to find and use existing word embeddings from online. You must use four different embeddings. We recommend using Word2Vec, GloVe, a pre-trained BERT and fine-tuned BERT model (for this one you should use: https://huggingface.co/google-bert/bert-base-uncased if possible). Though BERT must be two of your embeddings, you are free to choose one different than Word2Vec and GloVe for your other two embeddings. In **code/data.py** you will see example code for extracting token embeddings from text.

BERT For one BERT embedding, simply use the pre-trained model. For the fine-tuned, you are free to use your own techniques. Our baseline expectation/one good avenue would be to explore parameter-efficient fine-tuning methods like LORA (Low-Rank Adaptation). Please refer to the following link as to how to to use LORA with Huggingface `https://huggingface.com/docs/diffusers/en/training/lora`. In **code/fine_tune_bert.py** you will see some skeleton architecture to suggest possible starting points as you work to fine-tune the model.

Data Preprocessing

For any of these embeddings, you will find that the embedding dimension is not likely to match the measurement dimensions. Hence you will have to downsample from your embeddings (see [1] for details) and for that you can use the provided **downsample_word_vectors** from the **code/preprocessing.py** script.
Further, trim the 5 seconds and last 10 seconds of the output data to better match voxel measurements.
Finally you should use **make_delayed** from the **code/preprocessing.py** to insert delays from 1 through 4 seconds inclusively - explain what this does and why this might be useful.

Model and Evaluation Metrics

Now fit a linear model to each of the embeddings you generated in the previous step.

- Fit a ridge regression model, and report the mean correlation coefficient (CC) for different embeddings
- Devise a scheme to cross validate the different models and select the best performing linear regressor. Detail your procedure and report the mean test CC, median test CC, top 1 percentile CC and top 5 percentile CC at least. Look for other creative domain specific ways to evaluate performance.
- For the best embedding, perform a more detailed evaluation by examining the distribution of the CC across voxels. Plot the distribution. What do you notice?

Interpretation

Do a deep dive interpretation of your model in specific use-cases for the (fine-tuned) BERT model:

- For a given test story, identify the voxels where the model performs well.
- For these voxels, run SHAP (https://shap.readthedocs.io/en/latest/) and LIME to (https: //github.com/ma to identify influential words that strong affect the response. Why do we only do this for the voxels that we perform well for? Think about the "P" in PCS.
- Compare the words discovered by SHAP and LIME, and visualize them. Do these words make intuitive sense to you? How are they different? How are the words discovered different across voxels?
- Repeat this analysis for another test-story.

Stability Check

Stability concerns should be kept in mind throughout the analysis. However, explicitly make a section that is dedicated to analyzing in detail the stability of one of your choices or judgement calls (whether that be data cleaning/processing, hyper parameters, models, regularization, visualization choices or other). Thoughtfully and creatively choose a stability study that is not necessarily common place but would make you more confident in your final model in the context of your ultimate goal.

5 Note on Grading

As the final project is a group project, being a good collaborator on the project will be taken into account for each individual's grade on the final project (just as for lab4). After the project is submitted, we will send a Google form for each group member to evaluate the collaboration received from other members of their group. The final score for each student will be a combination of the student's collaboration score from their group mates, and the overall project score.

References

[1] Shailee Jain and Alexander Huth. "Incorporating Context into Language Encoding Models for fMRI". In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio et al. Vol. 31. Curran Associates, Inc., 2018. URL: https://proceedings.neurips.cc/paper_files/paper/2018/file/f471223d1a1614b58a7dc45c9d01df19-Paper.pdf.