Predicting Age from Social Media Language

Group 5: Angel Vasquez, Eliana Durell, Max Frohman

Project Recap

Research Question: What linguistic features can be leveraged to predict a writer's age or age group in an age prediction NLP task?

- Motivation:
  - Moderation and Safety
  - Marketing and Audience Targeting
  - Security and Threat Detection
  - Research and Data Labeling

Hypothesis: Using a transformer-based approach will pick up on both broader structural features (syntax) and more fine-grained age-indicating language usage like specific topics, punctuation trends, and slang

Keywords: age classification, text classification, social media, blogging

Problems

Do we have any problems that we faced?

- Transformer models are very large and slow to train
- Colab and other free resources are insufficient for training models
- Dataset size (600k+ samples) increases training time

Dataset: Blog Authorship Corpus

- From "Effects on Age and Gender on Blogging" by Schler et al., 2006
- Collection of over 600,000 posts from over 71,000 blogs on blogger.com as of 2004
- Age labels for posts spanning between 13 and 48

age	gender		
	female	male	Total
unknown	12287	12259	24546
13-17	6949	4120	11069
18-22	7393	7690	15083
23-27	4043	6062	10105
28-32	1686	3057	4743
33-37	860	1827	2687
38-42	374	819	1193
43-48	263	584	847
>48	314	906	1220
Total	34169	37324	71493

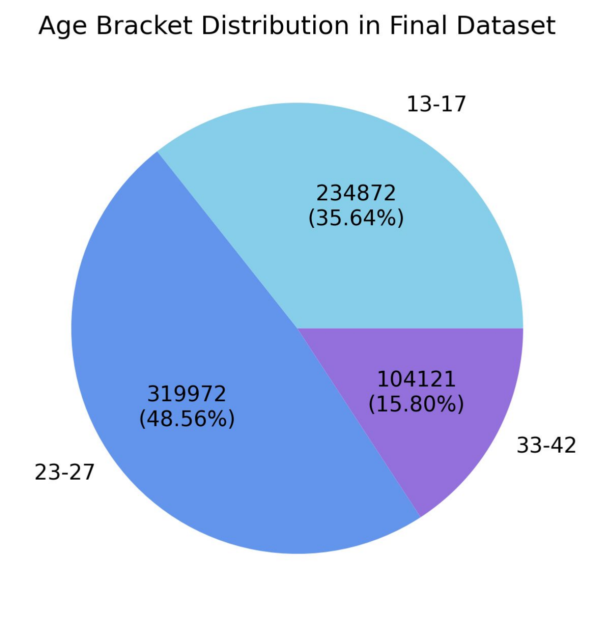
Table 1 Blogs Distribution over Age and Gender

Classed as →	10's	20's	30's
10's	7036	1027	177
20's	916	6326	844
30's	178	1465	1351

Table 7 Confusion matrix for the age classifier using all features

Dataset Concerns

- There does exist some bias towards the 23-27 age bracket
- The dataset is older
  - Data may contain older “slang” which may be an identifier for our model
  - May run into issues with modern trends and slang used by younger generations

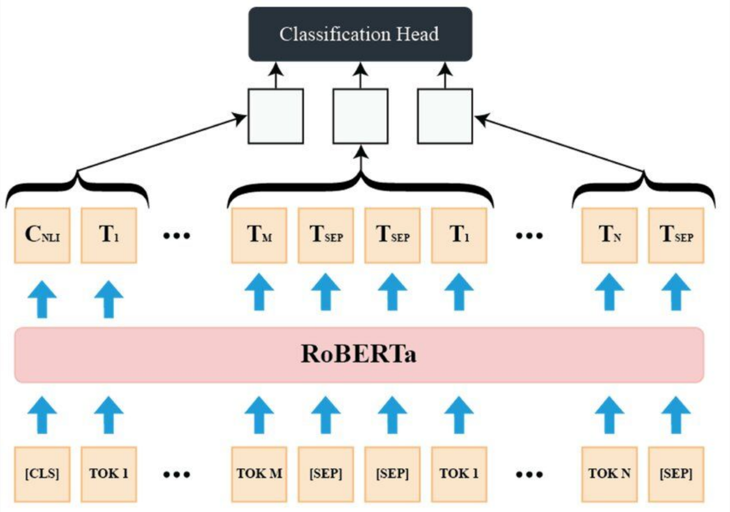


Tools

- Python libraries
  - PyTorch, pandas, NumPy, sklearn, matplotlib
- Hugging Face
  - Transformers, Datasets, Evaluate
  - RobertaForSequenceClassification
    - AutoTokenizer, Trainer and TrainingArguments, EarlyStoppingCallback
- LimeTextExplainer
- Lonestar6 supercomputer

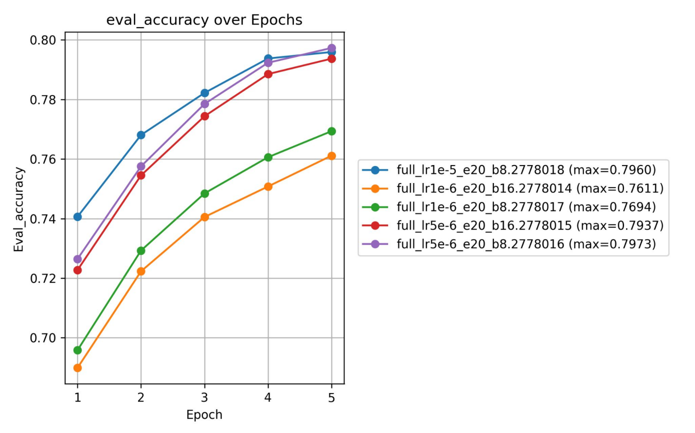
Model

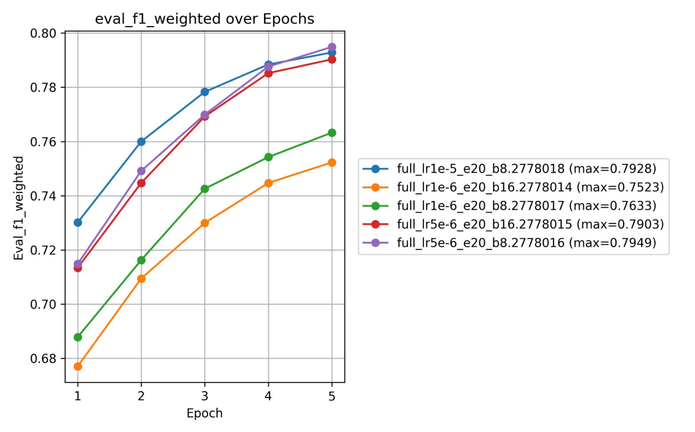
- RobertaForSequenceClassification
  - Model provided by the Hugging Face transformers API
  - RoBERTa with a sequence classification head
    - Additional head is specialized for working with sequences of text (like blog posts)



Model Training

- Currently testing different parameters to finetune our pretrained model
  - Current best results:
    - Batch size: 8
    - Learning rate: 5e-6
- Hugging Face API handles evaluation metrics
  - Loss, Accuracy, F1, Precision and Recall, Confusion Matrix





TACC Lonestar6

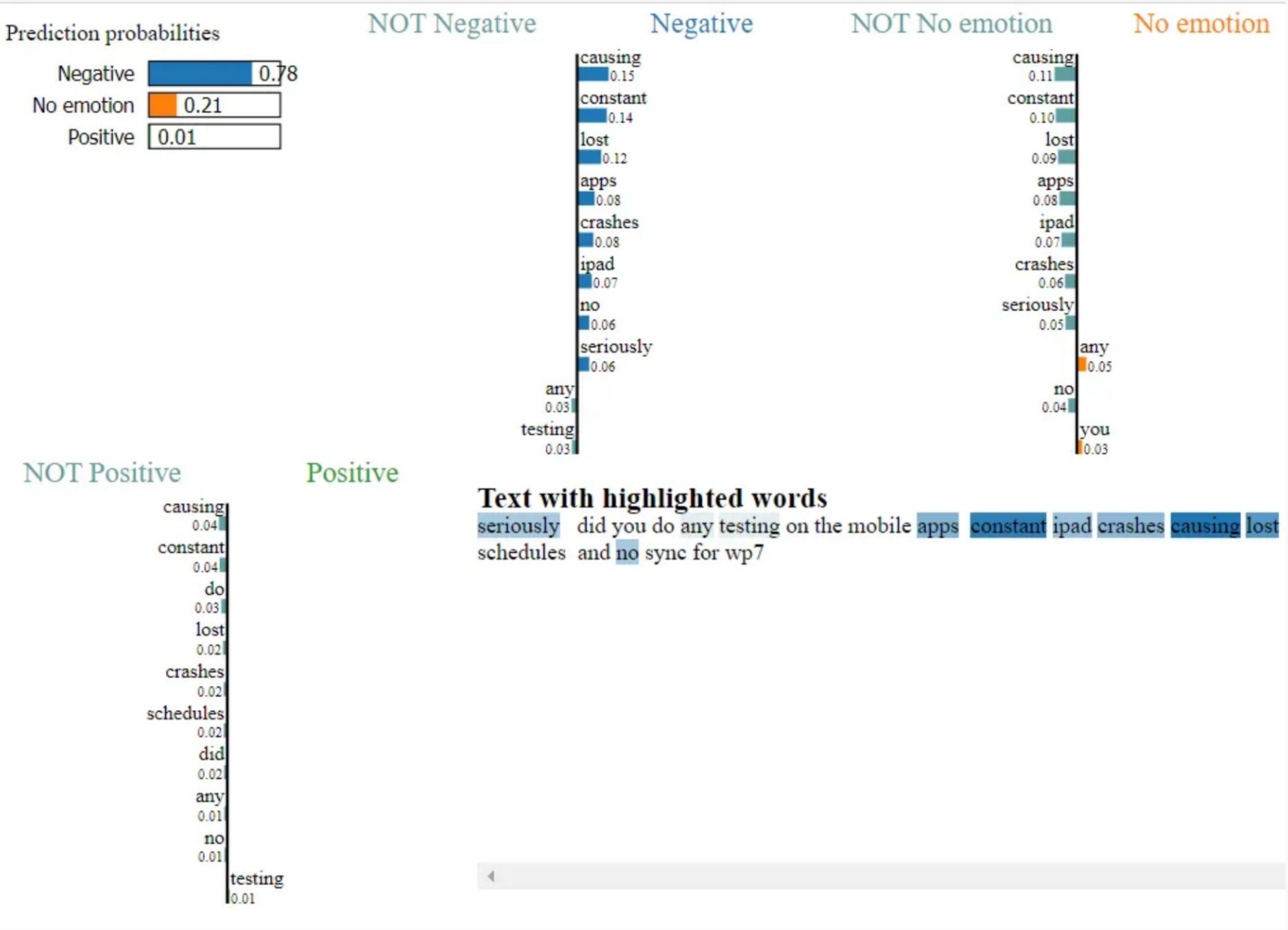
- The size of our transformer necessitates a high powered computing system for speedy training
- Allows for Python code to be run with NVIDIA A100 GPUs
- Scripts run with batch processing through the Simple Linux Utility for Resource Management (slurm)

```
#!/bin/bash
#SBATCH --job-name=full_lr1e-5_e20_b8
#SBATCH --partition=gpu-a100
#SBATCH --output=full_lr1e-5_e20_b8.%j.out
#SBATCH --error=full_lr1e-5_e20_b8.%j.err
#SBATCH --account=CCR24017
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=128
#SBATCH --time=20:00:00
#SBATCH --mail-type=ALL,TIME_LIMIT_50,TIME_LIMIT_90,TIME_LIMIT
#SBATCH --mail-user=mbf1102@rit.edu

set -e
cd $SLURM_SUBMIT_DIR

export TOKENIZERS_PARALLELISM=false

srun -N1 -n1 --exclusive bash -c "source /scratch/10746/maxfroh/ling581/envs/li
ng581/bin/activate && python /scratch/10746/maxfroh/ling581/ling581_final/train
er.py --num_epochs=20 --learning_rate=1e-5 --batch_size=8" &

wait
```

Testing

LIME (Local Interpretable Model-agnostic Explanations

- What - Tool that can provide explanations of predictions made by models
- Why - To better understand why a model is performing poorly
    - Able to see what parts of the text the model is paying attention to
- How
    - Treats the model as a black box and perturbs the inputted text
    - Fits a sparse linear model around the input
- Implementation
    - LimeTextExplainer and the names of the target classes
    - Function that takes in a list of strings and returns the prediction probabilities for each class
    - explain_instance
-



Prediction probabilities
Negative 0.78
No emotion 0.21
Positive 0.01
NOT Negative
Negative
causing 0.15
constant 0.14
lost 0.12
apps 0.08
crashes 0.08
ipad 0.07
no 0.06
seriously 0.06
any 0.03
testing 0.03
NOT No emotion
No emotion
causing 0.11
constant 0.10
lost 0.09
apps 0.08
ipad 0.07
crashes 0.06
seriously 0.05
any 0.05
no 0.04
you 0.03
NOT Positive
Positive
causing 0.04
constant 0.04
do 0.03
lost 0.02
crashes 0.02
schedules 0.02
did 0.02
any 0.01
no 0.01
testing 0.01
Text with highlighted words
seriously did you do any testing on the mobile apps constant ipad crashes causing lost schedules and no sync for wp7

Testing

Ablation Study

- What - an experiment to understand the importance of specific components (features) in a language model
- How does LIME fit in
- Strategies
    - Top-N
    - Progressive
    - Random

To-dos

What we have left to do

- Finalize model
- LIME and ablation test
- Find modern texts and see how the model performs

Questions?