Machine Learning Project Mid-term Report

Peizhong Gao
2023213526
Academic of Arts & Design
Tsinghua University
gpz23@mails.tsinghua.edu.cn

Yuzhe Zhang
2023215229
Global Innovation Exchange
Tsinghua University
zhang-yz23@mails.tsinghua.edu.cn

Matthew Taruno
2023280596
Global Innovation Exchange
Tsinghua University
wujiahui23@mails.tsinghua.edu.cn

1 Background

Writing is fundamental to your success and self-improvement. From just *the way you type and move your mouse*, could we predict how well you will score on a writing test? These insights might help you understand what writing behaviors lead to better writing performance. To collect data on the way humans type and move their mouse during a writing task, the exact environment works as follows. Participants of this project were hired from a crowdsourcing platform called Amazon Mechanical Turk to do an argumentative writing task. 2471 participants were asked to write an argumentative essay within 30 minutes in response to a writing prompt adapted from a retired SAT test taken by high school students. While participants took this writing test, a keystroke logging program written in JavaScript was embedded into the test-taking platform website. These events (median of 3000) fundamentally include when the key/mouse was pressed and released, action time, cursor positions, type of keyboard action, and the actual words typed. We believe this quantitative insight has high potential to be useful for understanding the psychological processes behind writing, nature of these writing tests that millions of children have to do every year, and maybe even insights on how you could be a better writer.

2 Progress

We consider feature engineering to be top priority. We have 12 core features: down time, up time, action time, position, word count, text change, activity, down_event, up_event, text_change, cursor_position, word_count. From these 12 features, we have created 350 features.

Recovering Original Essay. Our first area of focus is recovering the original essay based on the events. This step involves 'id', 'activity', 'cursor_position', and 'text_change' and based on the activities, reconstruct the original essay. From this recovery of the essay, we are able to create features specifically based on this essay content such as punctuation counts.

Categorized Count Features.

- Activity Counts: With 6 types of activities ('Input', 'Remove/Cut', 'Nonproduction', 'Replace', 'Paste', 'Move'), activity counts would generate 6 columns.
- Event Counts: If we assume a similar number of distinct 'down_event' and 'up_event' types (as not specified), and given the example includes 'q', 'Space', 'Backspace', etc., let's estimate about 20 different events for both 'down_event' and 'up_event', leading to 20 * 2 = 40 columns.

Preprint. Under review.

- Text Change Counts: Given the 'text_change' includes alphanumeric characters replaced by 'q' and other special characters, let's estimate around 20 different types of 'text_change', leading to 20 columns.

Gap-based features in time series data are a powerful tool for understanding temporal dynamics and patterns in user behavior. We implemented 8 different gaps (1, 2, 3, 5, 10, 20, 50, 100), and assuming 3 features ('action_time_gap', 'cursor_position_change', and 'word_count_change') for each gap, this leads to 8 * 3 = 24 columns. There gaps simply represent comparisons between an event and an event "gap" intervals ago, most simply. Here the short gaps (e.g. 1,2,3) help you understand rapid sequences of action, for example for the 'action_time' variable. Medium gaps (e.g. 5, 10, 20) help you understand short-term behavior patterns, and long gaps, like the difference in 'action_time' now and 100 keystrokes back, might encode broader patterns in behavior such as gradual slowdown over a long writing session or breaks taken between sections of the essay.

Statistical Summaries and Aggregation Features

We performed a comprehensive aggregation operation on the obtained features. They are grouped by the unique identifier `id` for each record. For this grouped data, we calculated a series of statistical aggregations on multiple columns: `down_time`, `up_time`, `action_time`, `cursor_position`, and `word_count`. These aggregations are intended to provide a multi-faceted understanding of the underlying patterns and variations.

The specific aggregations applied include: Mean, Standard Deviation, Minimum, Maximum, Last, First, Standard Error of Mean, Median and Sum. The exact feature and transformation set will be chosen from experimentation.

Typing Dynamics and Latency

Here, we get features such as sentence count (how many sentences the participant wrote), sentence length (mean, std, max, first, last, etc.), and sentence word_count. Paragraph features are essentially the same but for the paragraph level.

Ratio Features 4 columns for different ratios that encode information that might be useful to evaluate their scores:

- word time ratio $= \frac{wordcount_{max}}{uptime_{max}}$
- word event ratio $= \frac{wordcount_{max}}{eventid_{max}}$
- event time ratio $= \frac{eventid_{max}}{uptime_{max}}$
- idle time ratio $= \frac{actiontimegap1_{sum}}{uptime_{max}}$

For example, for idle time ratio, the higher the idle time, we might assume that the test taker might generally do a worse job.

3 Challenges

The difficulty of progress is to set up the experiments that would lead to higher scores in a scalable way. Currently, all our time is invested into gaining a strong understanding of the data and feature engineering. The biggest concern is we currently do not know how to improve the leaderboard performance, we may need to find more scalable and innovative methods.

Model based improvements and ensemble modeling can and will be experimented with for gains, however currently we have found that LightGBM performs strongly with a 0.61586 RMSE on training, and a submission score of 0.586 RMSE for the leaderboard, a ranking of 192 at the time of Nov 21 2023.

We have yet to try some of the exploratory ideas we proposed in the last checkpoint report, namely (1) involving external data and data augmentation. (2) Evaluation metric hacks for mean squared error (3) Transfer learning and knowledge distillation - perhaps based on emotion detection based on typing patterns. (4) Dimensionality reduction to potentially improve model speed.