# CAN WE KNOW HOW GOOD YOU WRITE BY HOW YOU TYPE?

# OUR TEAM

## Matthew Taruno

Global
Innovation
Exchange
Tsinghua
University

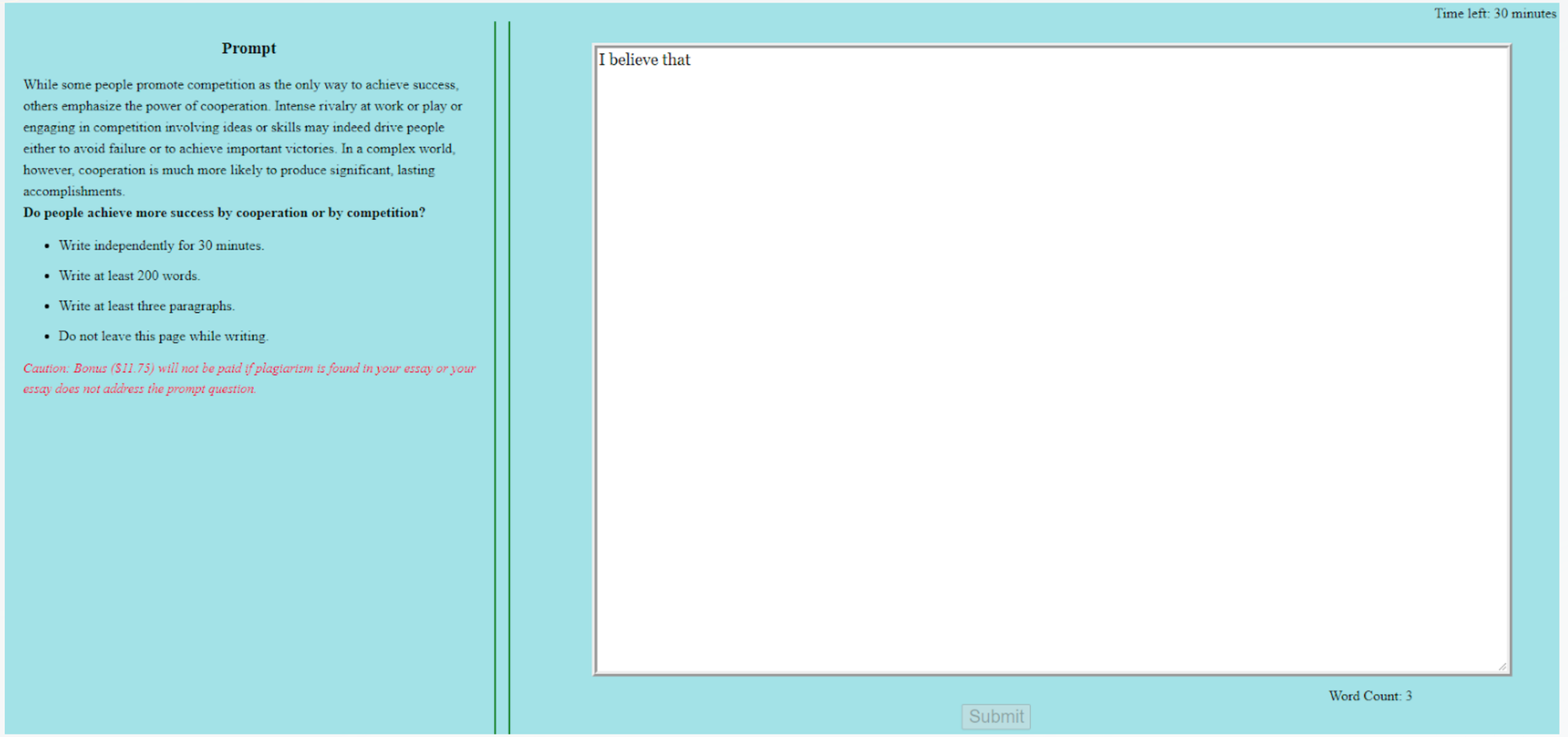
## Zhang Yuzhe

Global
Innovation
Exchange
Tsinghua
University

## Gao Peizhong

Academic of
Arts & Design
Tsinghua
University

# A WRITING TEST




## THOUGHT EXPERIMENTS

- faster typers score better?
- better to think longer before typing?
- best word length/sentence structure?
- best burst pattern?
- how engaged are test takers?

## PREDICT

writing quality (score 1-6)

# SERIOUSLY? KEYSTROKES?

*Results indicated that keystroke indices accounted for **76%** of the variance in essay quality and up to **38%** of the variance in the linguistic characteristics.*

**Table 3. Correlations between Essay Scores and Keystroke Indices**

| Keystroke Index | *r* | *p* |
|---|---|---|
| Verbosity | 0.819 | <.001 |
| Local Extremes (30s time window) | -0.476 | <.001 |
| Entropy (30s time window) | 0.472 | <.001 |
| Median Latency | -0.436 | <.001 |
| StdDev Events (30s time window) | 0.397 | <.001 |
| Largest Latency | -0.359 | <.001 |
| Backspaces | 0.308 | <.001 |
| StdDev Recurrence (30s time window) | -0.297 | = .001 |

A linear regression analysis was calculated with the eight keystroke indices as predictors of students' essay scores (score range: 1-6). This analysis yielded a significant model, $R^2 = .758$, $RMSE = 0.377$, $p < .001$, with three variables that combined to account for 76% of the variance in the essay scores: *Verbosity* [$\beta = 1.03$, $p < .001$], *Largest Latency* [$\beta = -.09$, $p < .001$], and *Backspaces* [$\beta = .39$, $p < .001$]. The follow-up ten-fold cross validation analysis produced a significant model with similar statistics, $R^2 = .737$, $RMSE = 0.386$.

Uncovering the Writing Process through Keystroke Analyses (International Conference for Data Mining 2016)

# BASE DATASET

*event_id*

*down_time* time (ms) when a key or the mouse was pressed

*up_time* release time of the event

*action_time* duration of the operation (i.e. up time - down time)

*activity* The category of activity which the event belongs to.

i.e. Nonproduction, Input, Remove/Cut, Paste, Replace, move from [x1,y1] to [x2,y2]

*down_event* The name of the event when the key/mouse is pressed

*up_event* The name of the event when the key/mouse is released

*text_change* The text that changed as a result of the event (if any)

*cursor_pos* The character index of the text cursor after the event

*word_count* The word count of the essay after the event

| Event ID | Down Time | Up Time | Action Time | Event | Position | Word Count | Text Change | Activity |
|---|---|---|---|---|---|---|---|---|
| 1 | 30185 | 30395 | 210 | Leftclick | 0 | 0 | NoChange | Nonproduction |
| 2 | 41006 | 41006 | 0 | Shift | 0 | 0 | NoChange | Nonproduction |
| 3 | 41264 | 41376 | 112 | I | 1 | 1 | I | Input |
| 4 | 41556 | 41646 | 90 | Space | 2 | 1 |  | Input |
| 5 | 41815 | 41893 | 78 | b | 3 | 2 | b | Input |
| 6 | 42018 | 42096 | 78 | e | 4 | 2 | e | Input |
| 7 | 42423 | 42501 | 78 | l | 5 | 2 | l | Input |
| 8 | 42670 | 42737 | 67 | i | 6 | 2 | i | Input |
| 9 | 42873 | 42951 | 78 | e | 7 | 2 | e | Input |
| 10 | 43041 | 43109 | 68 | v | 8 | 2 | v | Input |
| 11 | 43289 | 43378 | 89 | Space | 9 | 2 |  | Input |
| 12 | 44560 | 44605 | 45 | Backspace | 8 | 2 |  | Remove/Cut |
| 13 | 44661 | 44762 | 101 | e | 9 | 2 | e | Input |
| 14 | 44954 | 45032 | 78 | Space | 10 | 2 |  | Input |
| 15 | 45325 | 45381 | 56 | t | 11 | 3 | t | Input |
| 16 | 45460 | 45538 | 78 | h | 12 | 3 | h | Input |
| 17 | 45640 | 45730 | 90 | a | 13 | 3 | a | Input |
| 18 | 45741 | 45808 | 67 | t | 14 | 3 | t | Input |
| 19 | 45933 | 46011 | 78 | Space | 15 | 3 |  | Input |

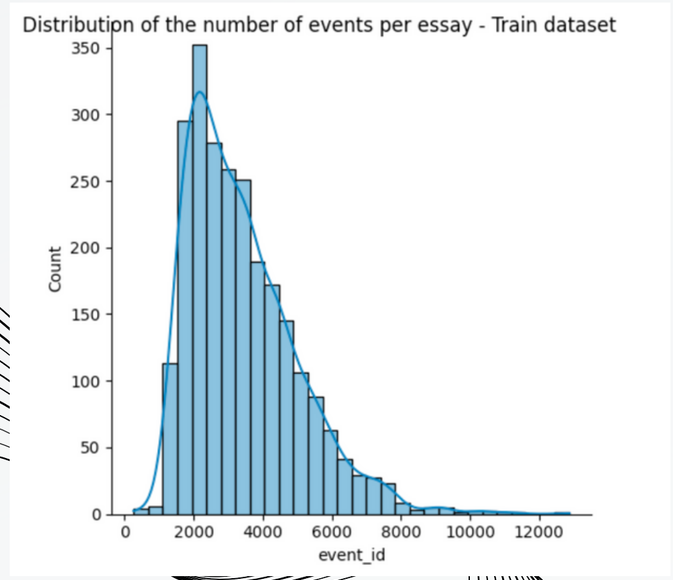
# INFRASTRUCTURE

actionListener()

mouse inputs

keystroke inputs

2471 participants



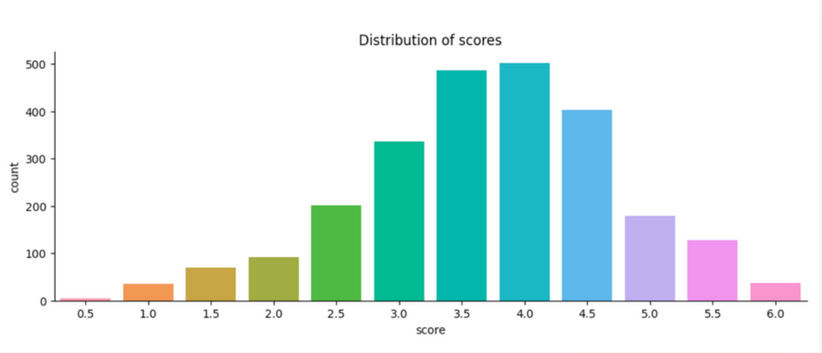

## LEVEL 0

- *event_id*
- *action_time*
- *up_time*
- *down_event*
- *up_event*
- *text_change*
- *activity*
- *cursor_pos*
- *word_count*

## LEVEL 1

- *time (gaps)*
- *cursor position*
- *word count*
- *statistical summaries*
- *activity counts*
- *event counts*
- *text change counts*
- *punctuation counts*
- *ratios*

396 features




## MODELS

- *LGBM*
- *XGBoost*
- *CatBoost*
- *SVM*

ENSEMBLE

PREDICT

writing quality (score 1-6)

# FEATURE ENGINEERING

*event_id* | *up_time* | *action_time* | *activity* | *down_event* | *up_event* | *text_change* | *cursor_pos* | *word_count*

## ESSAY RECONSTRUCTION

*recover original essay from:*

*id, activity, cursor position, text_change*

anonymized reconstruction:

*qqqqq qqq. qqq*

getEssays()

## AGGREGATIONS

*kurtosis*

*nunique*

*mean*

*skew*

also applied to the time gap features

shift entire ID aggregated dataset by lag amount

## TIME FEATURES

*lagging technique*

captures temporal nature and resolutions of the dataset

*gaps*

[1, 2, 3, 5, 10, 20, 50, 100]

immediate sequential dependencies

longer term patterns

*keystroke regularities*

*typing speed*

*ex. lagged uptime*

## LATENCY

*largest*

*smallest*

*median*

*initial pause*

*pause duration*

## RATIOS

- 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 $= \frac{actiontimegap1_{sum}}{uptime_{max}}$

## COUNTS

*activity*

*event*

*text change*

**396 features**

# FEATURE AGGREGATIONS

*event_id* *up_time* *action_time* *activity* *down_event* *up_event* *text_change* *cursor_pos* *word_count*

AGGREGATIONS

- *quantiles*
- *nunique*
- *mean + std*
- *skew + kurt*
- *min/max*
- *sum*

```
print("Engineering statistical summaries for features")
feats_stat = [
    ('event_id', ['max']),
    ('up_time', ['max']),
    ('action_time', ['max', 'min', 'mean', 'std', 'quantile', 'sem', 'sum', 'skew', kurtosis]),
    ('activity', ['nunique']),
    ('down_event', ['nunique']),
    ('up_event', ['nunique']),
    ('text_change', ['nunique']),
    ('cursor_position', ['nunique', 'max', 'quantile', 'sem', 'mean']),
    ('word_count', ['nunique', 'max', 'quantile', 'sem', 'mean'])]
for gap in self.gaps:
    feats_stat.extend([
        (f'action_time_gap{gap}', ['max', 'min', 'mean', 'std', 'quantile', 'sem', 'sum', 'skew', kurtosis]),
        (f'cursor_position_change{gap}', ['max', 'mean', 'std', 'quantile', 'sem', 'sum', 'skew', kurtosis]),
        (f'word_count_change{gap}', ['max', 'mean', 'std', 'quantile', 'sem', 'sum', 'skew', kurtosis])
    ])
```

# LATENCY

*event_id* *up_time* *action_time* *activity* *down_event* *up_event* *text_change* *cursor_pos* *word_count*

LATENCY

| | |
|---|---|
| *largest* | The longest delay between consecutive keystrokes for each writer. Indicates moments of deep thinking or hesitations. |
| *smallest* | The shortest delay, showing the fastest typing speed. |
| *median* | The median value of delays, offering insight into the typical typing rhythm. |
| *initial pause* | The time before the first keystroke, possibly reflecting preparation time. |
| *pause duration* | Counts of pauses in different duration ranges (half a second, one second, etc.), useful for understanding the frequency and length of breaks in typing, which might relate to cognitive processes like planning or revising text. |

# MODELLING

## LightGBM

- ***LightGBM*** *is an efficient machine learning framework based on* ***gradient boosting algorithms****, ideal for handling large data.*
- ***Tree-based approach****, capable of uncovering complex non-linear relationships.*
- *Parameters like* ***'num_leaves', 'max_depth'*** *are adjusted*

## CatBoost

- ***CatBoost*** *is a highly automated machine learning method, particularly adept at* ***handling categorical features.***
- *It* ***reduces overfitting*** *and* ***improves model generalization*** *through its unique algorithms.*
- *Parameters such as* ***'depth', 'iterations', and 'learning_rate'*** *are tuned*

## SVM

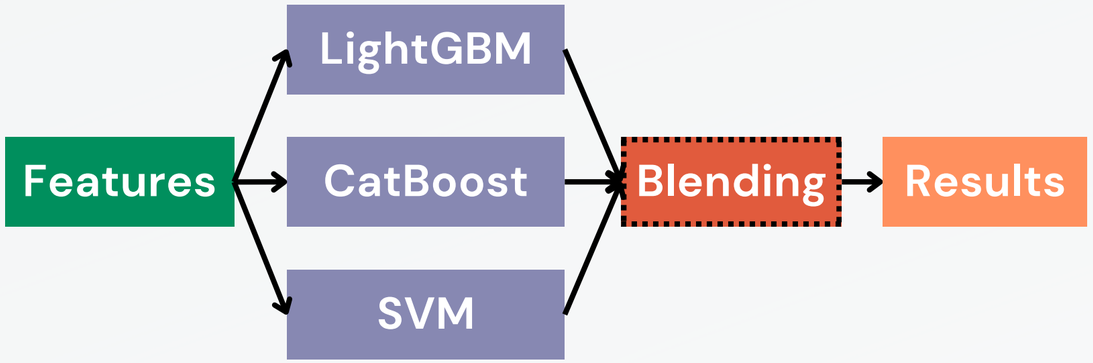
- ***Support Vector Machines (SVM)*** *are powerful classifiers suitable for small to medium-sized datasets.*
- *They work by finding the optimal boundary to differentiate between classes.*
- ***Kernel functions*** *like 'RBF', 'linear', or 'poly' are tested for better performance*

# EVALUATION AND RESULTS

## Evaluation Metric

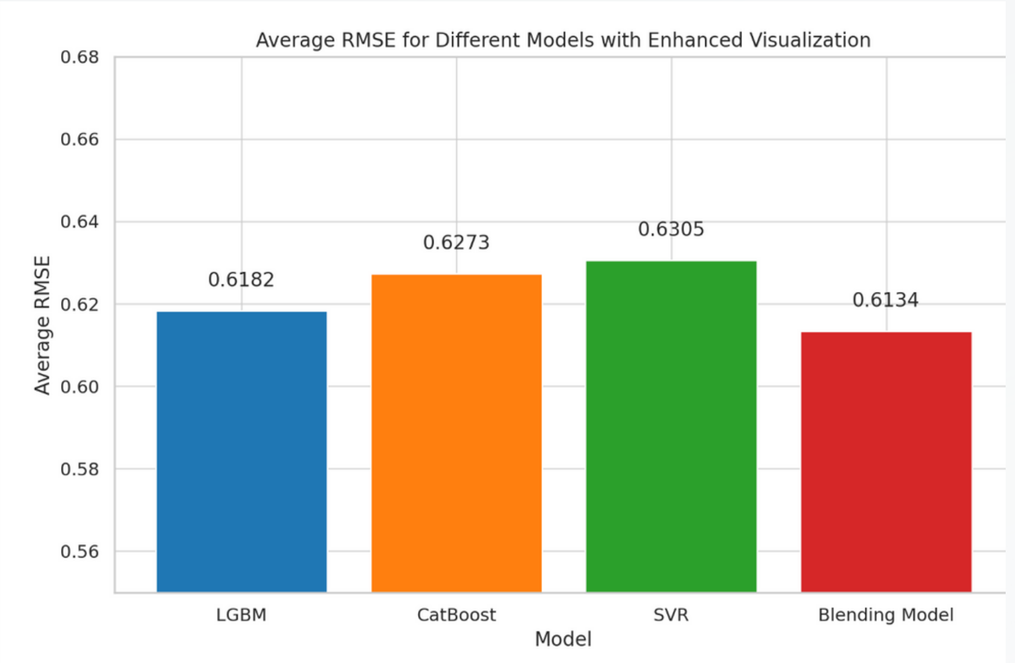
$$\text{RMSE} = \left( \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \right)^{1/2}$$

## Framework




***Best Blending_weights***:
*{'lgbm': 0.459, 'catboost': 0.246, 'svr': 0.295}*

## Verification




## Result

***Score on the hidden test dataset: 0.578***
***Rank: 20% (world record RMSE: 0.570)***

# THANK YOU