

# ERP 赛题说明

本赛题基于 P300 拼写范式，采集多人 EEG 数据集，旨在测试不同算法在无训练的条件下，识别 ERP 电位的性能。

## 一、实验范式

本赛题采用 P300 拼写范式。视觉刺激界面为如图 1 所示的 6×6 棋盘格排布（包括字母 A-Z、数字 1-9 及下划线），上方文本框显示需拼写的字符。实验中，刺激通过设置的二项式闪烁方式以随机顺序呈现，同时闪烁的字符被人脸图片覆盖（人脸由代码生成，非真人图像）。



图 1 视觉刺激界面

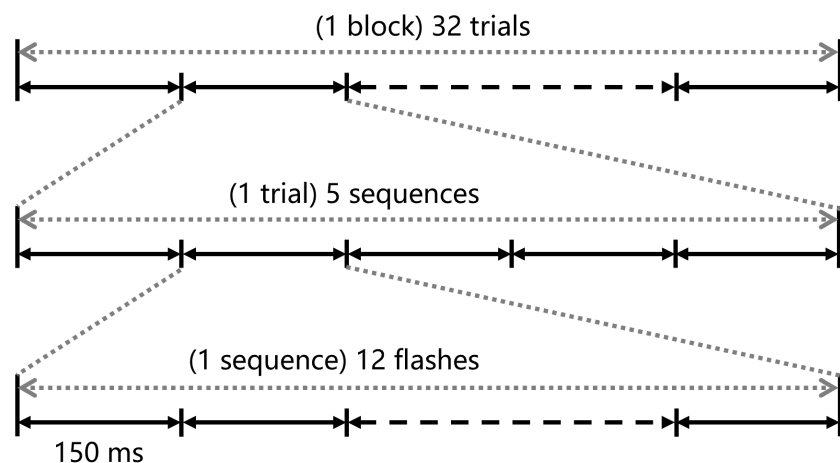


图 2 实验流程

在本次实验中，每位被试以 block 为单位采集数据，每位被试采集一次 block。

被试在一次 block 中被要求完成随机的 32 个字符的拼写，拼写一个字符作为一次 trial。每次 trial 开始前，屏幕上会出现红框提示待拼写字符，提示持续时间 1 秒。每次 trial 由 5 个随机序列闪烁（5 sequences）组成，闪烁序列间的间隔设置为 850 毫秒。每个随机闪烁序列包含 12 次闪烁（12 flashes），每次闪烁的字符依照本范式中定义的二项式规则，闪烁的呈现形式是人脸出现与消失各持续 75 毫秒，即每个 flash 持续 150 毫秒。被试视力或矫正视力正常，在实验过程中被要求集中注意力，刺激时注视提示的目标。整个实验流程如图 2 所示。参赛者需要通过分析给定被试的 EEG 信号，在线上测试中实现跨被试拼写字符预测。

## 二、实验数据

表 1 导联序号-名称对照表

导联序号	1	2	3	4	5	6	7	8
导联名称	Fpz	Fp1	Fp2	AF3	AF4	AF7	AF8	Fz
导联序号	9	10	11	12	13	14	15	16
导联名称	F1	F2	F3	F4	F5	F6	F7	F8
导联序号	17	18	19	20	21	22	23	24
导联名称	FCz	FC1	FC2	FC3	FC4	FC5	FC6	FT7
导联序号	25	26	27	28	29	30	31	32
导联名称	FT8	Cz	C1	C2	C3	C4	C5	C6
导联序号	33	34	35	36	37	38	39	40
导联名称	T7	T8	CP1	CP2	CP3	CP4	CP5	CP6
导联序号	41	42	43	44	45	46	47	48
导联名称	TP7	TP8	Pz	P3	P4	P5	P6	P7
导联序号	49	50	51	52	53	54	55	56
导联名称	P8	POz	PO3	PO4	PO5	PO6	PO7	PO8
导联序号	57	58	59	60	61	62	63	64
导联名称	Oz	O1	O2	ECG	HEOR	HEOL	VEOU	VEOL

表 2 Trigger 定义

定义	Trail 开始	Trial 结束	Sequence 结束	[A, B, C, D, E, F, 2]	[G, H, I, J, K, L, 9]	[M, N, O, P, Q, R, 8]
Trigger	101-136	241	200	1	2	3
定义	[A, S, T, U, V, W, X]	[B, H, Y, Z, 1, 3, 4]	[C, I, O, 5, 6, 7, _]	[D, J, P, V]	[E, K, Q, W, 3]	[F, L, R, X, 4, _]
Trigger	4	5	6	7	8	9
定义	[G, M, S, Y, 2, 5]	[N, T, Z, 6, 9]	[U, 1, 7, 8]			
Trigger	10	11	12			

\* Trail 开始的 Trigger (101-136) 分别对应 36 个字符, 表示该次 trial 的拼写目标, 也是本赛题中需要预测的字符标签

实验数据使用博睿康 64 通道脑电采集设备采集, 第 65 导联为 trigger 信息, 导联序号-名称对照表如表 1 所示。数据原始采样率为 1000 Hz, 降采样到 250 Hz, 未做其他滤波处理。具体 trigger 定义如表 2 所示, 前三项为 Trial 开始、Trial 结束及 Sequence 结束的 trigger, 第四项开始为每次 flash 的字符集合及其对应 trigger, 其余为系统预留。在线上测试时, 与类别信息相关的 Trial 开始 trigger 全部替换为 101, 仅用作同步。

数据流采用模拟在线方式提供。每调用一次数据读取方法, 可获得一个新数据包, 数据包中包含 40ms 的实验 EEG 数据(最后一个数据包长度可能小于 40ms), 以及在该数据包记录过程中收到的 trigger 信息。在同一 block 中, 数据包按照时间顺序依次发送。若测试数据中包含多组 block 数据, 则一组数据发送完毕后, 数据读取方法被再次调用时, 将会开始下一组数据的 EEG。而当所有实验数据发送完毕后, 程序终止标记 finishedFlag 将被置为 1。参赛算法检测到 finishedFlag 为 1 后, 需要结束 run()方法执行。

### 三、算法规范

参赛算法调用数据读取方法获取脑电数据。数据读取方法被调用一次, 比赛系统会返回一个新数据包, 参赛算法可以对新数据包进行缓存并处理。当算法认为接收到的数据足以满足判决条件时, 需要调用反馈方法向比赛系统报告识别结果。比赛系统根据返回结果时计数的 sequence 个数计算有效时间, 并结合反馈正确率, 综合计算出模拟信息传输速率。

参赛算法需要同时满足以下几个约束条件:

#### 1、试次起止约束:

在对单一试次数据的检测识别过程中, 参赛算法需要在接收到该试次 trigger 之后开始检测, 并且最迟在接收到下一试次 trigger 时进行反馈报告。否则, 报

告结果将被错判为后续试次的识别结果。

### 2、单试次最大数据长度约束：

从试次开始的 **trigger** 信号开始，参赛算法给出反馈结果使用的数据长度不应超过试次结束的 **trigger** 信号，否则该试次识别结果将被视为无效，该试次结果为 0。不同被试，不同试次的检测数据长度可以不同。

### 3、算法终止约束：

当接收到数据包中 **Endflag = 1** 时，意味着所有实验数据均已发送完毕，参赛算法需要停止处理并自行退出。

## 四、赛题框架

### 1、参赛者用例

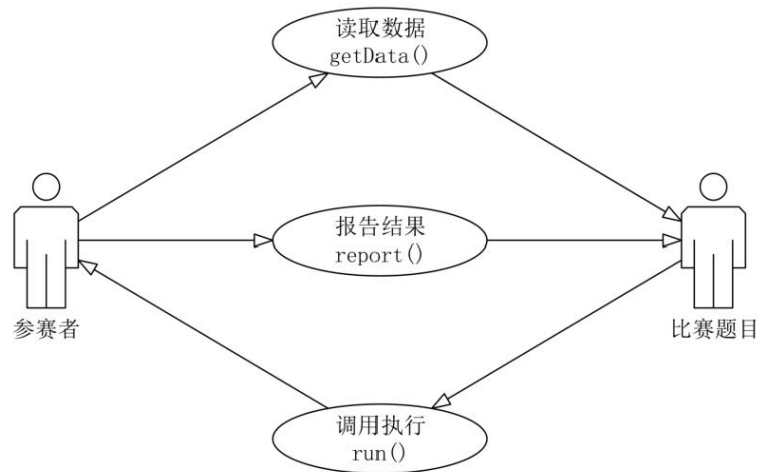


图 3 参赛者用例

### 2、系统主体框架图

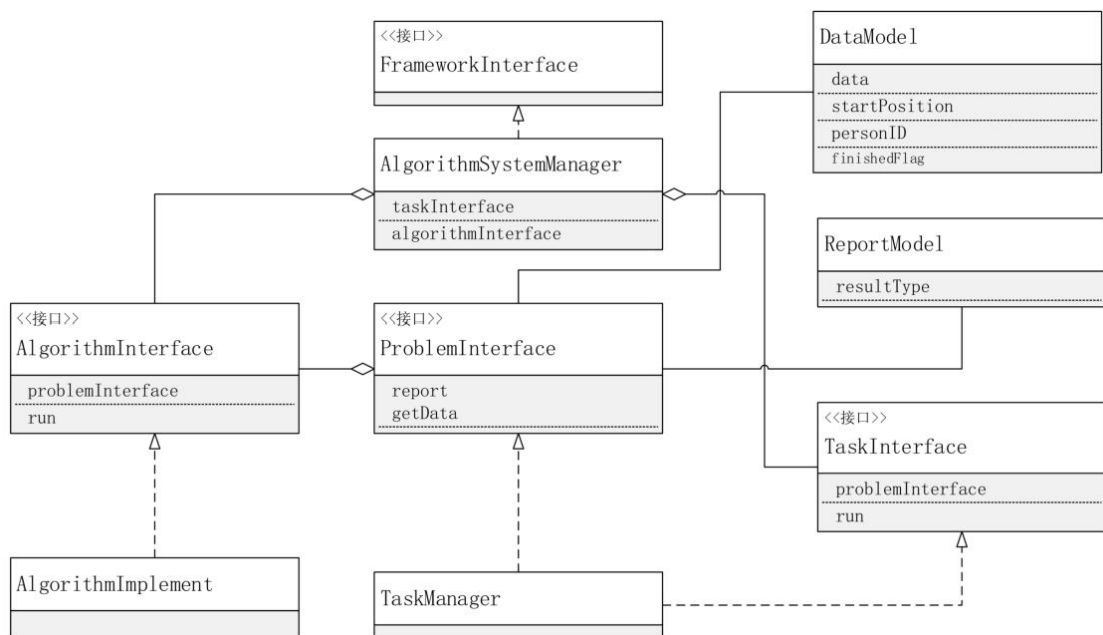


图 4 系统主体框架

(1) FrameworkInterface 框架接口

该接口主要负责赛题程序与外部执行系统的连接。该框架接口的实现类为 AlgorithmSystemManager，实现接口定义的所有函数。

(2) ProblemInterface 题目接口

该接口是面向参赛者的赛题接口，主要负责题目与参赛算法之间的数据传递及结果报告。参赛者可以通过该接口获取比赛数据，并通过该接口报告识别结果。比赛题目需要根据参赛算法获取比赛数据的长度，以及报告结果的正确率综合给出比赛评分。

(3) TaskInterface 赛题接口

该接口主要负责实现赛题的数据填充、获取得分、清除数据以及清除报告结果，框架通过该接口实现对赛题的调用。

(4) AlgorithmInterface 算法接口

通过该接口比赛题目可以对参赛算法进行验证计算。参赛者需要实现该接口。在执行过程中，参赛算法需要通过 ProblemInterface 接口获取数据，并且通过该接口报告结果。同时，参赛者需要控制算法的计算复杂度，否则当运行时间超过预定长度时，系统将自动终止该计算进程，所获成绩无效。

### 3、数据模型

(1) DataModel 参赛者数据模型

1) data: float 类型矩阵，分段数据。例如包含有 64 导 EEG 数据+1 导 trigger 信号，在 1000Hz 采样率下，以 40ms 对数据分段，则单次获取的 data 为 65\*40 个点。

2) startPosition: int 类型标量，当前分段数据起始时刻相对于该 block 数据起始时刻的索引位置。

3) personID: int 类型标量，当前数据来源受试者序号。

4) finishedFlag: bool 类型标量，测试结束标志。当参赛算法通过 get\_data() 函数获取数据包中该字段为 True 时，需要自行退出程序运行。

### 4、参赛者相关接口函数

(1) ProblemInterface

该接口由出题方负责实现，包括数据获取方法及结果反馈方法。在算法运行前，该接口的实现类会被注入参赛算法实现类中。算法执行过程中，可以调用该接口获取数据，并通过结果反馈方法报告识别结果。出题方根据数据获取方法的被调用次数，及结果反馈的正确性进行综合评分。

1) def get\_data(self):

输入参数：无

输出参数：DataModel

实现功能：获取下一分段实验数据。

2) def report(self, reportModel):

输入参数：ReportModel

输出参数：无

实现功能：反馈识别结果。

(2) AlgorithmInterface

参赛者需要将程序运行过程填入 run 函数中。在算法执行过程中，通过 ProblemInterface 接口 get\_data 方法获取 DataModel 类型数据，并通过 report 方法返回 ReportModel 类型结果。当通过 get\_data 获取的 DataModel 数据中 finishedFlag 为 true 时，意味着数据处理完毕，该函数需要自行退出运行。

1) def run(self):

输入参数：无

输出参数：无

实现功能：算法分析过程。

提交格式

本赛题程序使用 python 语言编写，需提交基于 python 3.9 版本的扩展名为.pyc 的文件。

## 5、提交样例

参考配套代码。

参赛者可通过修改 Algorithm 文件夹中的代码完成算法，为了避免未知错误，请勿在主目录内添加文件夹。完成后重新打包程序（包含 AlgorithmImplement 文件夹和 config.toml）--> 分组 --> 具体分组 --> 计算单元 --> 定义计算单元 --> 上传程序包 --> 提交到比赛 --> 选择比赛 --> 部署 -->完成比赛。

部署完成后在赛题的排行榜中查看比赛成绩；

需要注意的是，为防止参赛者修改代码框架作弊，保护评分程序会完全覆盖参赛者的代码（除了 AlgorithmImplement 目录和 config.toml）在提交到比赛 --> 部署时，启动的实际为 评分程序 + 参赛者的 AlgorithmImplement 目录，其余运行配套代码均为服务器内置程序（包括 main.py 等文件，服务器内置评分程序与范例中程序框架基本相同，但包含评分功能和读取服务器比赛数据功能）。

## 6、评分方式

本赛题以模拟信息传输速率作为评分标准：

$$ITR = \frac{60}{T} (\log_2 N + P \log_2 P + (1 - P) \log_2 \frac{1 - P}{N - 1})$$

其中， $T$ 表示输出字符所需时间， $N$ 表示可选目标数（ $N = 36$ ）， $P$ 表示字符识别正确率。 $ITR$ 的单位是 bits/min。参赛算法的最终得分为被试平均 ITR，需要特别指出的是，当单个被试的拼写准确率不超过 0.5 时，该被试的字符识别准确率将会被置零；此外，本系统中 $ITR$ 是按照理想 $ITR$ 进行计算，即输出字符所需时间不包含模拟休息时长，因此 $T$ 可以表示为：

$$T = mT_s$$

其中， $m$ 表示参赛算法反馈一个字符结果使用的序列个数， $T_s$ 为一个常数，表示理想的序列时长。

## 7、性能评估方法

参赛算法通过数据读取方法获取新数据包。当所得数据包内含有 trigger 信号时，评分系统将自动开始记录算法识别过程中所使用数据包的个数，直至反馈方法被调用。单一试次共有 5 个序列，按时间顺序发送，单个序列读取 3 个及以上 flash 的 trigger 时，参赛算法使用的序列计数加一，记录的使用序列数总是为 1-5 间的整数。从 trigger 开始到反馈方法被调用时所获取的序列计数将用于计算该试次的模拟时长。而平均准确率将根据算法反馈结果与真实刺激的一致性进行计算。

需要特别指出，在本比赛项目中每一个包含 trigger 的数据包，其依然被视为是前一试次的数据。而新试次数据是从包含 trigger 数据包的下一个数据包开始计算。因此参赛算法不可在获取到包含 trigger 信号的数据包时立刻反馈，而最早需获取到下一数据包后才可反馈。

## 8、结果反馈异常处理

### 1) 重复多次报告

在一个试次时间内，参赛算法多次反馈结果，则仅按照第一次反馈的时间及结果进行记录。

### 2) 结果未反馈

若在一个试次时间内，参赛算法未反馈结果，则该试次时长将被记录为 9 秒，同时判决结果将被记录为误判。

### 3) 结果反馈超时

当参赛算法反馈结果时超出本次 trial 结束的 trigger 信号，则判决结果将被记录为误判，同时该结果反馈时长被设置为 9 秒。

### 4) 算法执行超时

为满足脑-机接口系统实时处理需求，本项目同时对参赛算法的计算复杂度有一定要求。本比赛项目将会根据比赛数据量大小确定一个计算时间。若算法复杂度过高导致系统运行超时，则该算法比赛成绩将被视为无效。