# 12月7日组会报告（第一次）

通过意念书写实现高效的脑电文本转换

普适与混合智能实验室-侵入式脑机接口研究-段裕

## 背景介绍

**（1）脑机接口的基本概念**

脑机接口是一个使大脑能够与外部设备直接交流的系统，它是一门研究、测绘、帮助、增强或修复人体的认知或感觉运动的学科。脑机接口BCI(Brain Computer Interface)这一名词最早出现在Jacques Vidal教授在1973年发表的Toward Direct Brain-Computer Communication这一论文中。

**（2）脑机接口的工作流程**

典型的BCI系统工作流程分为三步，第一步是采集脑电信号（或其他大脑活动信号），接着对这些信号进行处理，例如去噪、数字化等等，最后对这些信号进行解码或者映射以控制外部设备或者还原并输出（例如显示文字、发出声音等）

**（3）脑机接口技术分类**

按照信号采集方法的不同，脑机接口分为非侵入式脑机接口、半侵入式脑机接口、侵入式脑机接口，它们分别在头皮处、脑膜间、大脑皮质内采集信号，技术风险依次递增，但能够获取的信号质量依次递增。

**（4）脑-文本应用研究现状**

这一研究主要是让瘫痪患者恢复与外界沟通的能力，已有的技术方法是使用非侵入式BCI技术中的**EEG**，也即头皮脑电图谱技术（其他的非侵入式BCI技术还包括MEG（脑磁图谱技术）、PET（正电子发射断层扫描技术，向血液中注射放射性示踪剂追踪大脑葡萄糖消耗情况）、fMRI（功能性核磁共振成像，检测大脑耗氧量分布）、fNIRS(功能性近红外光谱,通过检测血红蛋白的磁性得到大脑含氧量分布)。这一技术通过电极阵列采集头皮处的脑电信号来对大脑活动进行解码或者映射，通过稳态视觉诱发电位SSVEP或者P300事件相关电位REP来实现文字输入，前者是利用不同频率的信号刺激视网膜引发不同的电位从而实现映射，后者则是利用目标刺激出现后约300ms出现的正向电位丰实现映射。

通过**SSVEP**稳态视觉诱发电位进行文本输入的过过程如下：受试者面前有一个显示屏，上面显示了6x6的字符阵列—包括26个小写字母和10个数字，这些字符会以固定的频率闪烁，各个字符闪烁频率不同，用户想要输入哪个字符就盯着哪个字符看，此时大脑的早期视觉区域会产生趋于该刺激频率的EEG信号，将这样的信号进行采集去噪解码和对比，就能知道用户盯着的是哪一个字符。

通过**P300**事件相关电位进行文本输入的过程如下：与前者相同仍然显示一个6x6的字符阵列，但是字符不会闪烁，而是随机的按行列高亮，当用户盯住的字符所在行或者列被高亮时，就可以检测到事件相关电位并进行对比和输出。

总结起来，这两种主流的非侵入式脑机接口文本输入技术都是对神经信号进行对比和映射，其中由于P300是随机高亮的故而速度较SSVEP更慢，但由于它们均采用非侵入式的办法，采集的脑电信号噪声多、有效信息少，因而平均每分钟准确输入的字符数较低。同时观察实验的流程，这样的输入方式十分不自然，过于机械。

本文采用侵入式的办法在受试者的大脑皮层下植入了两个96电极的微电极阵列，通过对受试者在想象手写文字时的脑电活动进行解码，使用RNN模型将其实时地翻译为文字，在线的打字速度达到了90字符每分钟，实时准确率为94.1%，在离线状态下，借助于通用语言模型进行校正，其准确率达到了99%以上。

## 主要成果和发现

### 神经活动强烈可重复

使用主成分分析将记录的神经活动(multiunit threshold crossing rates，多阈值跨越率)降低到包含最大方差的前3个维度（图1-b）。

### 书写每个字符的潜在神经活动模式显著可区分

通过调整神经活动的时间来消除书写速度的反复变化（图1-c）。在C上方的插图中，示例时间翘曲函数显示为字母“m”，并且相对接近于恒等线(每次试验的翘曲函数用不同颜色的线绘制)。

### 大脑神经活动编码了笔尖的运动，包括笔尖的速度

预期的2D笔尖速度通过交叉验证从神经活动中线性解码(每个字符都被显示出来)。解码后的速度在整个试验中平均，并进行整合以计算笔轨迹(橙色圆圈表示轨迹的开始)

### 同一字符或相似字符的神经活动具备极好的聚簇性

使用t-SNE非线性降维方法将每次实验中记录的神经活动数据降至二维。（图1-e）这一可视化图像表明了同一字符或相似字符的神经活动具备极好的聚簇性。

### 短时复杂移动的解码要从根本上比点到点的直线运动更容易解码

研究人员分析了16个手写字符(持续1秒)和16个手写直线运动(持续0.6秒)对应神经活动的时空模式。字符和直线的空间维数相似，但字符的时间维数高两倍，表明更复杂的时序模式构成最近邻距离的增加和更好的分类性能。

总结：在意念书写的过程中，瘫痪数年的受试者T5的运动神经活动表达仍然强烈并且可重复，同时针对于不同的字母书写，神经活动表达是显著可区分的，对于同一字符或相似字符的神经活动具备良好的聚簇性。实验还表明这一神经活动编码了想象中的笔尖运动轨迹和笔尖移动速度。最后给出时域上的复杂性使得诸如手写的短时复杂移动的解码要从根本上比点到点的直线运动更容易解码。

## 实验过程

### 总体说明

实验总体目标时训练一个能够解码手写字符神经活动的实时RNN模型，基本的策略是使用**停等-执行范式**，也即在T5尝试逐字符的在脑海中完成句子书写后会有一个短暂的休息时间。实验按天进行，它将每一天作为一个**会话session**，会话持续3到5小时，每一个会话包含多个5到10分钟的**块block**，块内的实验是连续不间断的，其中的神经活动数据会被记录下来。

在每个会话中都包含句子书写和字符书写两个部分，一共进行了11组实验。在字符书写实验中，开始会有一个停等阶段，在该阶段屏幕上出现一个字符，提示方块为红色表示等待输入，接着红色方块变为绿色，表示执行阶段开始，此时T5应立即尝试在脑海中书写屏幕上的字母，这将持续1秒，紧接着就是下一个停等周期。注意在字符实验中不会应用实时解码器。

在句子书写实验中，每次试验开始时都有5秒的停等阶段，在此期间，屏幕上一个红色方块上方会出现一个句子(或问题)。接着红色方块变成绿色指示T5立即开始尝试写句子。当T5确定他写完了，他把头转向右边。我们的系统自动检测到右转头部，并使用这个信号触发下一次试验。

有四种类型的句子写作试验:

1. 句子复制(最常见的)。

2. 包含停顿的句子复制。在句子中随机插入符号#，表示T5在遇到这样的字符时稍作停顿(大约1秒钟)。

3. 根据记忆书写短语。在停等阶段，T5需要记住屏幕上的短语（不超过4个单词）。在执行阶段，短语消失，他必须凭记忆将短语书写出来。

4. 自由回答问题。T5需要自己组织语言回答屏幕上出现的问题。许多测试在开始提示后都有很长的停顿，因为T5用这段时间来思考问题的答案，并在开始写之前写下他的回答。

最后三种类型的测试只发生在两个“自由回答”环节，这两个环节的设计是为了评估RNN解码器在自由组合句子上的表现。自由回答会话的训练数据被设计为在T5的书写中创造更多的可变性，试图使RNN解码器对停顿和书写速度的变化更健壮。

## 语料选择

每一个提词打字评估实验中都会用到5个训练数据块，每一块包含10条语句，这些句子在实验前被收集，它们来自于英国国家语料库(BNC，British National Corpus)。首先，我们从BNC中最常见的2000个单词列表中随机选择单词。然后，对于每个随机选择的单词，BNC搜索包含该单词的例句。从这些例子中，我们手工选择了长度合理的句子(不超过120个字符)，其意思不会脱离上下文太混乱，以免分散T5的注意力。最终的结果是来自许多不同语境(英语口语、小说、非小说、新闻等)的不同句子样本。

在最后两组会话中，句子中随机加入了一些#字符表示停顿，T5拼写时遇到这一字符需要稍作停顿，这使得RNN模型适应不规则的书写速度和自由打字过程中可能经常出现的不可预测的停顿。

### 数据预处理

规定时间步，计算阈值交叉率

对于每一个电极，其记录的原始神经信号为一个电压时间序列，使用-4.5RMS（RMS为序列的均方根）作为阈值进行锋值检测，平均阈值跨越率不小于2Hz（也即平均1s内，神经信号波形曲线跨越这一阈值至少2次）的电极被认为具有锋值活动，该电极记录数据有效。

规定时间步长为10ms（数据分析）或20ms（RNN模型训练），将这一时间步内的阈值跨越次数进行记录，这样的阈值跨越次数在时间步、电极（192）、X三个维度张成的三维张量就构成了模型的原始的基本的输入，其中X维度在不同类型的实验中有不同的定义。

在字符拼写实验中，维度X被定义为每一个字符的实验编号，长度27，也即对于同一字符的意念书写实验会进行27次，注意同一字符的实验不会在连续的时间步上进行，由于有多个字符（32），因此这样的三维张量输入有32个，另外时间步维度长为201。

在句子拼写实验中，X维度被定义为句子，长度为句子数量例如102，由于句子书写时间很长，因此时间步维度长度也会更长（取所有句子中完成时间最长的），为10669，电极仍然是192个。

### Time-Warped PCA

这一步是对所有会话中字符数据输入进行时间对齐，首先对这些数据按块进行均值归一化处理，接着使用一维的高斯卷积核在时间维度上进行高斯平滑，最后调用TWPCA包得到对齐后的数据并将其存储

### HMMs Labeling

在进行数据训练时，我们无法实时的了解T5正在描画的字符是哪一个，因此需要使用隐马尔可夫模型进行数据标记，这里是指书写句子的实验，因为在这一类型的实验中会用到实时解码器，因此标注的数据输入即为数据预处理部分提到的三维张量。

主要用到的函数包括

1. labelDataset——该函数返回句子中每一个字符的起始时间步号letterStarts,和书写字符的持续的时间步数letterDurations，它们都是N×200的矩阵，其中N代表句子数量

2. constructRNNTargets——该函数返回可用于RNN监督训练的标记数据。包括*charStartTarget*，一个N×T的矩阵，其中T表示时间步，这一矩阵的值为0或者1，1标注了句子中包含字符书写锋值活动的时间步；

*charProbTarget，*一个N×T×31的独热编码矩阵，它指示了在书写某一句子出现字符书写锋值的时间步对应的字符是哪一个，用一个31维的独热向量表示；

*ignoreErrorHere*：一个N×T的矩阵，标注了哪些时间步的句子或字符是不被添加进训练数据的，这主要出现在自由问答环节，例如T5回忆不起来了，又或者没法回答屏幕的问题等等。

### Synthetic Data Augmentation

在上一步中获得了每一个句子中每一个字符对应的起始时间步与持续时间步数，因此可以把每一个字符从信号流中分割出来，并组织成一个样本数据库，此后，可以从库中随机挑选一串字符数据用于训练。这其中产生的数据约有110G