**比赛背景**

本次比赛的目标是将学生写作中的争论元素分类为“有效”、“充分”或“无效”。来自本次比赛的模型将有助于为学生获得关于他们的议论文的更多反馈铺平道路。借助自动指导，学生可以完成更多作业，最终成为更自信、更熟练的作家。

这里提供的数据集包含美国 6-12 年级学生撰写的议论文。这些文章由专家评分者注释，用于议论文中常见的话语元素：

* 引导——以统计、引文、描述或其他一些手段开始的介绍，以吸引读者的注意力并指向论文
* 立场- 对主要问题的意见或结论
* 主张- 支持该立场的主张
* Counterclaim - 反驳另一项主张或对该立场提出相反理由的主张
* 反驳- 反驳反诉的主张
* 证据- 支持主张、反诉或反驳的想法或例子。
* 结论性声明- 重申声明的结论性声明

你的任务是预测每个话语元素的质量等级。人类读者将每个修辞或论证元素按质量递增的顺序评为以下之一：

* 无效
* 足够的
* 有效的
* train.csv - Contains the annotated discourse elements for all essays in the test set.
  + discourse\_id - ID code for discourse element
  + essay\_id - ID code for essay response. This ID code corresponds to the name of the full-text file in the train/ folder.
  + discourse\_text - Text of discourse element.
  + discourse\_type - Class label of discourse element.
  + discourse\_type\_num - Enumerated class label of discourse element .
  + discourse\_effectiveness - Quality rating of discourse element, the target.

Try different ways of processing and creating our input:

* Include the full essay data (might be tough for some models with limited context length)
* Try different separators, order of dataset

Try different models:

* Bigger models
* Other BERT-like models
* Are there any pretrained models relevant to this dataset available?

Use previous competition data and insights (<https://www.kaggle.com/competitions/feedback-prize-2021/>)

Hope this helps you get started with this competition!

NOTE: Thie notebook was heavily inspired by Jeremy Howard's [amazing](https://www.kaggle.com/code/jhoward/iterate-like-a-grandmaster) [notebooks](https://www.kaggle.com/code/jhoward/getting-started-with-nlp-for-absolute-beginners). Be sure to check them out!

看以往的比赛代码！

目前的优化方法：

模型训练 epoch、学习率等

模型融合：样本多样性：将数据集切分分组。模型多样性：训练过程融合、训练结果融合（加权平均）

数据集增强

对抗学习

Cocolm

FGM

EMA

AutoModelForSequenceClassification.from\_config(cfg)

不同层学习率

Now：模型层、model.zerograd、FGM a 、数据集增强伪标签、text

Loss观察：sf2loss epoch=3，loss还可能继续下降

Lr2、lr3 loss epoch第3时微量上升

base\_fold5\_nopool\_lr2\_e4 valid loss在epoch=2时最小之后一直上升，但测试loss=0.639

提前停止

测试ema、rdrop

再跑一次mp large

跑一次roberta large