1
大家好，我们是张浩楠、蔡宇生和吴昊原。我们的工作是灾难侦测。我们的工作包括写提案。我们去认真清洗数据、设计模型去为了取得好成绩。我们努力编写这个展示的 ppt 和手稿。

2
这次展示的内容包括介绍数据、介绍模型、和提交。

3
我们的任务是灾难侦测。随着互联网的进步，更多的人选择去在互联网分享它们的故事。它们也可能在互联网上描述他们见到的灾难。那么，高层可以通过监听互联网去侦测灾难，而不是等市民去呼叫夭夭九，那么可能可以更早发现事故，派消防车等救援工作人员去支援，从而可以减小损失。

4
我们的任务是预测一个帖子是否跟真实灾难有关。

5
我们的训练数据跟测试数据的规模分别是 7613 和 3263。它们共同优的特征包括了编号位置、关键字和文本。训练数据有额外的标签。其中编号是每个样本有的唯一的值；位置是帖子发布的位置；关键字为帖子的关键字、文本时帖子的内容；标签代表这个帖子是否跟真实灾难有关，0 是无关，1 是有关。

8
在训练数据中，关键字和地点两组特种中，数据的缺失值比例分别为 0.8%和 33%。测试数据中也有相近的缺失值比例。我们曾经通过运用在这两个特征上独热编码后再线性回归方法去完成这个任务，结果是不好的。

9
帖子的文字没有缺失值。我也曾经研究过帖子文字的小特征，例如帖子的长度、帖子有多少链接、和帖子有多少井号。是没用的。

10
标签的 0 和 1 分别表示与真实灾难无关和有关。占比分别是 0.4 和 0.6。

11
接着我们应该处理文字，来使帖子文字的分布相近。我们的处理包括 1 删除不常见的符号、拆开略缩、删除链接、删除井号和删除重复的样本。

15
处理好数据后，我们可以开始设计模型。我们的设计使在 bert 的最后添加上一个分类器，最终得到规模为 2 的向量，对应着帖子与真实灾难有关和无关。

16
我们选择 bert-base-uncased 作为模型的谷关，bert-base-uncase 是大的，不区分大小写的。

它有 12 层变形金刚编码器 Transformer encoder 和 12 个注意力头 Attention head 和一亿一千万参数 parameter。

17
从抱脸 hugging_face 上下载预训练模型后，我们有令牌器 tokenizer 和 bert。令牌器首先在句子的头和尾添加特殊令牌[CLS]和[SEP]，分别表示句子的开头和结尾。

18
接着令牌器会切割词，切割的方案是子词令牌化，就比如像构词法拆开单词。Tokenizer 会被拆成两个，token 和井井 izer。

19
最后单词会变成数字序列 Sequence。一个单词将会对应一个数字，比如单词[CLS]将会对应数字 101，[SEP]对应 102，它不会对应 103 或其他数字，词嵌入的规模是 768。

20
那么我们最后就从一个句子得到一个数字序列。

21
Bert 的组成包括词嵌入 word embeddings 和 12 层的变形金刚编码器 Transformer encoder。

22
词嵌入将会被分成令牌嵌入 token embedding，段嵌入 segment embedding 和位置嵌入 position embedding。其中令牌嵌入中，一个数字对应一个令牌嵌入；段嵌入帮助区分单词是属于第一个句子还是第二个句子，在这个任务中，我们只输入一个句子，不需要第二个句子，全部的段嵌入将会是 0；位置嵌入能代表一个单词在句子的哪个位置，解决了传统的变形金刚中不能区分词的位置的问题。最后，词嵌入的规模将会是[n_tokens, 768]。

23
接下来是变形金刚编码器。

24
再接下来是 bert 的输出。我们的设计是我们拿到[CLS]上对应的嵌入，它规模是 768，其他的就没用了。我们再在后面接上密集层和 Softmax 激活函数，输出规模为 2 的向量，对应真灾难和假灾难。

25
我们冻结词嵌入器 word embedding 和变形金刚编码器 transformer encoder，只去优化密集层分类器。

26
我们的选择网格搜索的模型选择。我们的网格的参数是 1 层或 2 层或 3 层的密集层的层数量，和亚当优化器 Adam 和随机梯度下降优化器。

27
最后我们选择 1 层或 2 层的亚当优化器 Adam。它们是最快最准。它们的验证分数都很高。F1_score 0.83。值得注意的是，竞赛的评价指标是 F1_score，它代表着精确率 accuracy 和召回率 recall。越接近 1，精确率和召回率高。

28
得到两个模型的最好的参数后，我们再来交叉验证 kfold。我们分别创造 5 个实例。将数据集分为 5 个大小相同的自己，每轮选择 4 份作为训练集，剩下的 1 份做测试集。去分别训练 5 个模型，5 轮之后，选择验证分数最高的一个。

29
这是我们的交叉验证中的验证分数。一层密集层亚当优化器中的 5 个模型中，最优的是第 4 个，它的验证分数是 0.84。第二个种的最优的验证分数是 0.85。那么我们挑选出它们，我们有两个模型。

30
最后我们准备提交。我们提交选择软投票的集成学习。我们首先用两个模型预测测试数据的标签。那么就得到两个预测。每个预测中，每一行代表这个帖子分别属于与真实灾难无关和有关的概率，并且它们的加和应该是 1，由于我们的激活函数是 softmax。

31
我们把对应的概率求出均值，那么就得到投票的结果。我们选择每行的最大的一个中对应的标签。

32
那么就可以提交了。

33
我们的排行榜分数是 F1score 0.81，与之前的验证分数相近。

3435
这就是我们的展示。