

# 深度学习实验六

1190200708 熊峰

2022 年 6 月 8 日

目录	2
----	---

## 目录

<b>1 环境配置</b>	<b>3</b>
1.1 硬件配置 .....	3
1.2 软件配置 .....	3
<b>2 实验选题</b>	<b>3</b>
<b>3 数据</b>	<b>3</b>
3.1 训练数据 .....	3
3.2 测试数据 .....	4
<b>4 解决方案</b>	<b>5</b>
4.1 数据处理 .....	5
4.2 相关优化 .....	6
4.3 模型选择 .....	6
<b>5 实验结果</b>	<b>7</b>

## 1 环境配置

### 1.1 硬件配置

CPU : Intel(R) Xeon(R) Silver 4214

GPU : TITAN RTX 24G

MEM : 128G RAM

### 1.2 软件配置

OS : Ubuntu 20.04.1 LTS

PyTorch : Stable 1.11.0 CUDA 11.3

IDE : PyCharm 2021.3.2

## 2 实验选题

实验选题为 kaggle 上 Featured Code Competition, 题目为 **U.S. Patent Phrase to Phrase Matching**.

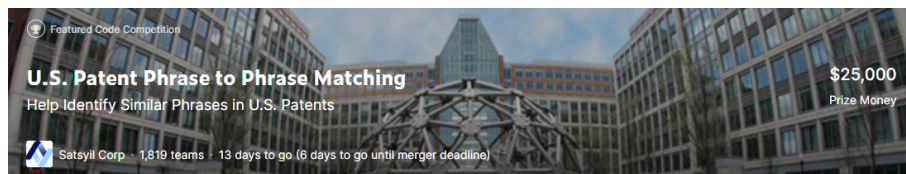


图 1: Theme

## 3 数据

### 3.1 训练数据

训练数据如下图所示, 对于给定的 anchor 和 target, 给出其相匹配的程度, 例如, 如果一项发明声称是“电视机”, 而先前的出版物描述了“电视机”, 那么理想情况下, 模型会识别出这两者是相同的, 并帮助专利律师或审查员检索相关文件。

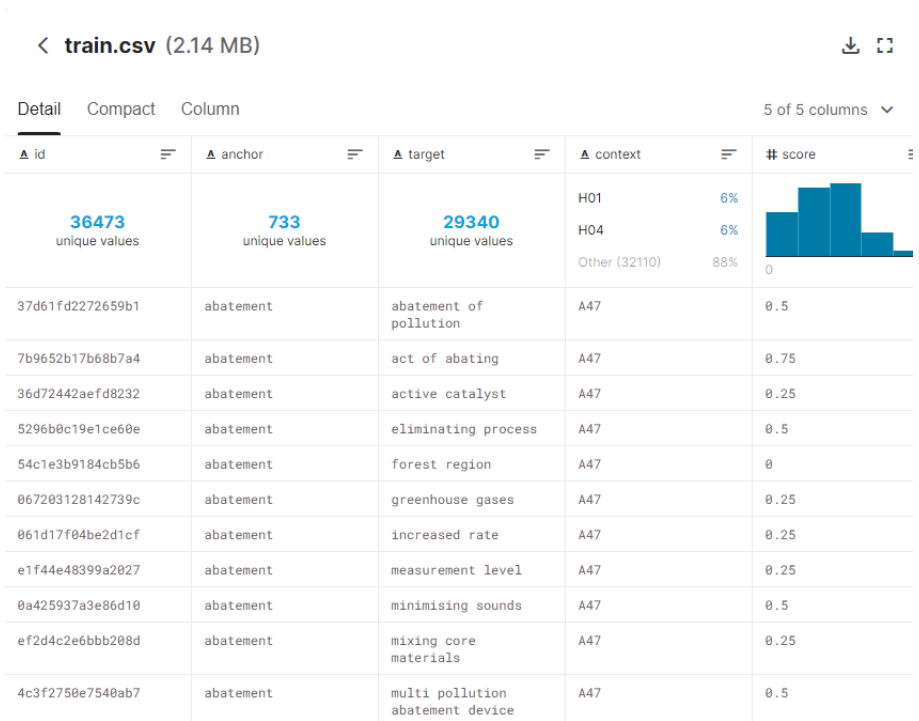


图 2: Train Data

3.2 测试数据

测试数据如下图所示，需要给出 anchor 和 target 的相关程度，并给出了相关的上下文的符号。

< test.csv (1.97 kB)

Detail

Compact

Column

4 of 4 columns

id	anchor	target	context
36 unique values	el display hybrid bearing Other (32) 6% 6% 89%	36 unique values	G02 B60 Other (31) 8% 6% 86%
4112d61851461f60	opc drum	inorganic photoconductor drum	G02
09e418c93a776564	adjust gas flow	altering gas flow	F23
36baf228038e314b	lower trunnion	lower locating	B60
1f37ead645e7f8c8	cap component	upper portion	D06
71a5b6ad068d531f	neural stimulation	artificial neural network	H04
474c874d0c07bd21	dry corn	dry corn starch	C12
442c114ed5c4e3c9	tunneling capacitor	capacitor housing	G11
b8ae62ea5e1d8bdb	angular contact bearing	contact therapy radiation	B23
faaddaf8fcb8a8a3f	produce liquid hydrocarbons	produce a treated stream	C10
ae0262c02566d2ce	diesel fuel tank	diesel fuel tanks	F02

图 3: Test Data

## 4 解决方案

### 4.1 数据处理

将 test 中的 anchor 和 target 连接到 Cooperative Patent Classification Codes Meaning 数据中，获取相关的上下文信息，并构建 dataset。

采样数据的分布，使用 DeBerta 分词器对待分类的句子分词，分词后的数据分布如下：

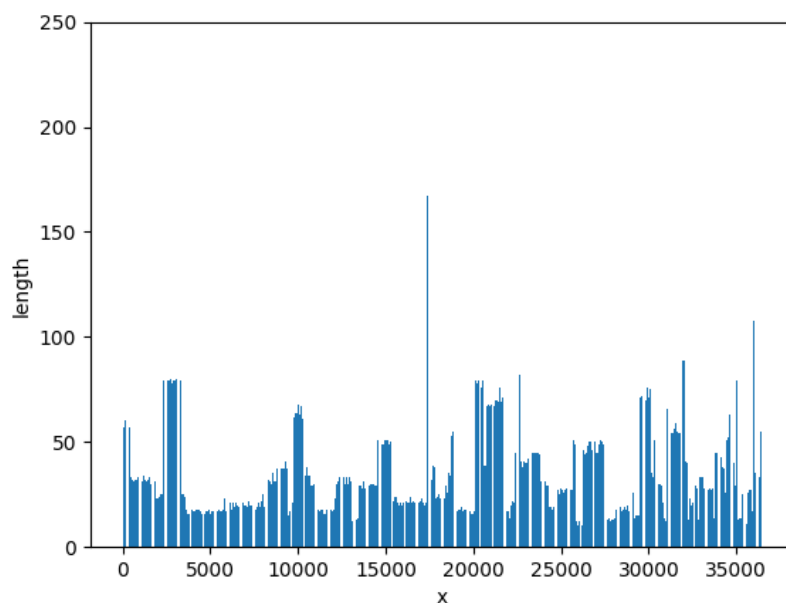


图 4: Statics

在后续处理中，将句子截断为 100，对不足 100 的补齐。

## 4.2 相关优化

使用 K 折交叉验证对训练过程优化，实验中选取 K 为 5。

## 4.3 模型选择

模型主要选用 DeBerta[1]，它主要提出了两点改进：解耦注意力机制和增强的 mask 解码器。

解耦注意力: 与 BERT 不同，在 BERT 中，输入层中的每个单词都使用一个向量来表示，该向量是其单词 (内容) 嵌入和位置嵌入的总和，单词间的权重分别根据其内容和相对位置使用解耦的矩阵进行计算。这是由于观察到一个单词对的注意力权重不仅取决于它们的内容，而且还取决于它们的相对位置。例如，单词 “deep” 和 “learning” 相邻时的关系要强于出现在不同句子中。

增强的 mask 解码器: 与 BERT 一样, DeBERTa 也使用 mask 语言模型 (MLM) 进行了预训练。MLM 是一项填空任务, 在该任务中, 模型被教导要使用 mask token 周围的单词来预测 mask 的单词应该是什么。DeBERTa 将上下文的内容和位置信息用于 MLM。解耦注意力机制已经考虑了上下文词的内容和相对位置, 但没有考虑这些词的绝对位置, 这在很多情况下对于预测至关重要。考虑一下句子 “a new store opened beside the new mall”, 其斜体字 “store” 和 “mall” 被 mask 以进行预测。尽管两个单词的局部上下文相似, 但是它们在句子中扮演的句法作用不同。(这里, 句子的主题是 “store” 而不是 “mall”。) 这些句法上的细微差别在很大程度上取决于单词在句子中的绝对位置, 因此考虑单词在语言模型中的绝对位置是很重要的。DeBERTa 在 softmax 层之前合并了绝对单词位置嵌入, 在该模型中, 模型根据词内容和位置的聚合上下文嵌入对 Masked 单词进行解码。

## 5 实验结果

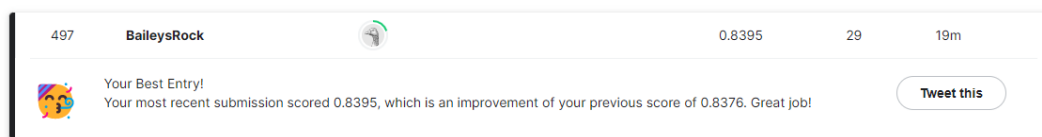


图 5: Score

在实验中先后尝试了多种优化, 最高得分为 0.8395, 排名 497/1819.

## 参考文献

- [1] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*, 2020.