分词器(tokenizer)

BPE分词器训练和使用示例代码(内含详细注释)

定义和概念

分词器将文本拆分成一个个词元(被称为token ’通常是比一个字或单词更小的单元) ’然后将每个token映射到一个唯一的数字ID °这样 ’文本就可以被转换成数字序列 ’可以被大语言模型所使用了 °

一般情况下 ’我们可以使用预训练的分词器 ’例如BERT `GPT `百度 `千问等都提供了分词器 °但是 ’在某些情况下 ’我们需要为特定任务定制分词器 °例如 ’如果我们正在处理一个特定的领域或语言 ’我们可能需要创建一个专门针对该领域的分词器 °或者针对特定任务 ’我们不需要分词器中的所有词汇 ’就可以创建一个只包含我们需要的词的自定义分词器 °

常用分词器

在Hugging Face的Transformers库中 ’models 模块提供了几种不同的模型用于分词 °基于这些模型和训练文本数据(比如预训练数据) ’我们可以训练用于特定任务的分词器 °

常见的分词器模型有 ：

- BPE(Byte-Pair Encoding) ：用于创建子词单位(subword units)的分词器 ’是最常用的模型之一 ；
- WordPiece ：类似于BPE ’但使用了稍微不同的方法来选择要合并的符号对 ’是最常用的模型之一 ；
- Unigram ：一种基于概率的方法 ’可以视为是WordPiece的泛化 ；
- CharBPETokenizer ：一种基于字符级别的BPE ’适用于需要更细颗粒度处理的语言 °比如形态丰富的土耳其语 °

分词器训练原理 ：以BPE为例

BPE分词器的训练过程如下:

1. 初始化 ：字符级token(词元)的拆分

BPE算法以单个字符作为初始词汇表的基础°如 ’对于英文文本 ’初始词汇表价格包含所有英文字母(大小写) `数字 `标点符号以及其他特殊字符等 °这可以确保即使在训练文本数据中从未见过的单词 ’也可以通过词元的组合来表示 °比如 ’我们自创一个英文单词"tiaoyu" ’它至少可以被拆分为"t" `"i" `"a" `"o" `"y" `"u" ’而这些字母都是初始词汇表中的词元 °

2. 词频统计 ：识别出最常出现的词元对
   BPE算法会遍历整个训练文本数据 ’并记录每一对相邻词元(之前迭代形成的词元)同时出现的次数 °这一步的目的是识别出最常共同出现的词元对 ’它们可能在后续步骤中并合并为一个新的词元 °
3. 合并词元 ：扩充词汇表
   选择最高频次的词元对 ’将它们合并成一个新的词元 °例如 ’在英文文本中 ’"t"和"h"可能是最常见的词元对之一 ’它们会合并成一个新的词元"th" °更新词汇表 ’加入新的词元(只增不减 ’之前的词元仍然保留) °
4. 重复迭代 ：直到词汇表大小满足要求
   重复步骤2和3 ’直到词汇表的大小达到预设的阈值(vocab_size)或满足其他停止条件(如 ’最小频次min_frequency等等) °如果设置了多个停止条件 ’通常会选择最先满足的条件作为停止迭代的依据 °比如 ’如果min_frequency较大的值 ’那最终生成的词汇表大小可能小于预设的vocab_size ’因为这时已经找不到高于min_frequency的词元对可以合并了 °不断执行上述过程 ’词汇表将从仅包含单个字符的初始词汇表扩展到更复杂的词元结构 °

在文本处理时 ’BPE会优先使用较长的字词来分割文本 °