视听信息系统导论第三次编程作业报告

目录

1 实验任务

1.1 数据准备

1.1.1 TASK 1

补全 dataset.py 中的 prepare_data 函数，并详细说明整个函数是如何构建词表的。

函数代码如下：

```
def prepare_data(data_path):
    with open(data_path) as f:
        data = f.read()

    ################################################################################
    # TODO: insert '#' at the end of each paragraph, delete all special chars; #
    #  you can leverage data.replace() and data.split() to accomplish that. #
    #  two or three lines of code should be sufficient.                 #
    ################################################################################
    # *****START OF YOUR CODE (DO NOT DELETE/MODIFY THIS LINE)*****

    data = data.replace('\n\n', '#')
    data = ' '.join(data.split())

    # *****END OF YOUR CODE (DO NOT DELETE/MODIFY THIS LINE)*****
    ################################################################################
    #                           END OF YOUR CODE                                   #
    ################################################################################
```

其中词表的构建流程是：

首先对数据进行预处理。

再提取唯一字符集合：`unique_chars = set(data)`。

将每个字符映射为一个唯一索引：`voc2ind[char] = idx`。

再创建反向映射：`ind2voc = {val: key for key, val in voc2ind.items()}`。

将 data 按 80% 和 20% 分割成训练集和测试集，使用`voc2ind[char]`将训练集和测试集中的每个字符转换为对应的整数索引。

将训练集和测试集的整数表示、字符到索引的正向映射、字符到索引的反向映射保存下来。

1.1.2 TASK 2

请完成 dataset.py 中的 HarryPotterDataset 类。(完成代码即可，不用在报告中写文字说明)

代码如下：

```
class HarryPotterDataset(torch.utils.data.Dataset):
    def __init__(self, data_file, sequence_length, batch_size):
        super(HarryPotterDataset, self).__init__()

        self.sequence_length = sequence_length
        self.batch_size = batch_size
```

```
        self.vocab = Vocabulary(data_file)

        with open(data_file, 'rb') as data_pkl:
            dataset = pickle.load(data_pkl)

        self.tokens = dataset['tokens']
        self.voc2ind = dataset['voc2ind']
        self.ind2voc = dataset['ind2voc']

        self.data = None
        self.sequences_in_batch = None
        ################################################################################
        # TODO: split self.tokens to len(self.tokens)//batch_size chunks, store the #
        #  reshaped data (and convert it to torch.LongTensor) in self.data; #
        #  Then compute how many sequences are there in each chunk, store that in #
        #  self.sequences_in_batch                                                 #
        ################################################################################
        # *****START OF YOUR CODE (DO NOT DELETE/MODIFY THIS LINE)*****

        self.data = torch.LongTensor(self.tokens[:len(self.tokens) // self.batch_size * self.batch_size])
        self.data = self.data.view(self.batch_size, -1)
        self.sequences_in_batch = self.data.size(1) // self.sequence_length

        # *****END OF YOUR CODE (DO NOT DELETE/MODIFY THIS LINE)*****
        ################################################################################
        #                            END OF YOUR CODE                                  #
        ################################################################################

    def __len__(self):
        ################################################################################
        # TODO: return the total number of sequences in dataset                        #
        ################################################################################
        # *****START OF YOUR CODE (DO NOT DELETE/MODIFY THIS LINE)*****

        return self.sequences_in_batch * self.batch_size

        # *****END OF YOUR CODE (DO NOT DELETE/MODIFY THIS LINE)*****
        ################################################################################
        #                            END OF YOUR CODE                                  #
        ################################################################################

    def __getitem__(self, idx):
        data = None
        ################################################################################
        # TODO: Based on idx, determine the chunk idx and the sequence idx of the chunk#
        #  fetch that sequence data from self.data and store that in data variable; #
        #  Note the data length should be sequence_length + 1                          #
        ################################################################################
        # *****START OF YOUR CODE (DO NOT DELETE/MODIFY THIS LINE)*****

        chunk_idx = idx // self.sequences_in_batch
        seq_idx = idx % self.sequences_in_batch
        start = seq_idx * self.sequence_length
        end = start + self.sequence_length + 1
```

```
        data = self.data[chunk_idx, start:end]

        # *****END OF YOUR CODE (DO NOT DELETE/MODIFY THIS LINE)*****
        ################################################################################
        #                           END OF YOUR CODE                                   #
        ################################################################################

        # returns input data and label data (next token of input) with their length sequence_length
        return data[:-1], data[1:]

    def vocab_size(self):
        return len(self.vocab)
```

1.2 构建模型

1.2.1 TASK 3

请完成 model.py 中的 HarryPotterTransformer 类。(完成代码即可，不用在报告中写文字说明）请基于构建的网络完成训练，绘制训练、测试的损失曲线和测试的准确率曲线。(请绘制在报告中)

代码如下:

```
class HarryPotterTransformer(nn.Module):
    def __init__(self, vocab_size, feature_size, num_heads):
        super(HarryPotterTransformer, self).__init__()
        self.vocab_size = vocab_size
        self.feature_size = feature_size
        self.num_heads = num_heads
        self.best_accuracy = -1

        self.embedding = None
        self.transformer_encoder = None
        self.decoder = None
        self.pos_encoding = None # you can omit this for Task 4

        ################################################################################
        # TODO: define the network                                                     #
        ################################################################################
        # *****START OF YOUR CODE (DO NOT DELETE/MODIFY THIS LINE)*****

        self.embedding = nn.Embedding(vocab_size, feature_size)
        encoder_layer = nn.TransformerEncoderLayer(
            d_model=feature_size,
            nhead=num_heads,
            dim_feedforward=4 * feature_size,
            dropout=0.1
        )
        self.transformer_encoder = nn.TransformerEncoder(encoder_layer, num_layers=2)
        self.decoder = nn.Linear(feature_size, vocab_size)
        self.pos_encoding = PositionalEncoding(feature_size)

        # *****END OF YOUR CODE (DO NOT DELETE/MODIFY THIS LINE)*****
        ################################################################################
```

```
        #                           END OF YOUR CODE                           #
        ########################################################################

    def forward(self, x):
        attn_mask = None # you can omit this for Task 4 and Task 5
        ########################################################################
        # TODO: finish the forward pass                                        #
        ########################################################################
        # *****START OF YOUR CODE (DO NOT DELETE/MODIFY THIS LINE)*****

        x = self.embedding(x) # [batch_size, seq_len, feature_size]
        x = x.permute(1, 0, 2) # [seq_len, batch_size, feature_size]
        x = self.transformer_encoder(x, mask=attn_mask) # [seq_len, batch_size, feature_size]
        x = x.permute(1, 0, 2) # [batch_size, seq_len, feature_size]
        x = self.decoder(x) # [batch_size, seq_len, vocab_size]

        # *****END OF YOUR CODE (DO NOT DELETE/MODIFY THIS LINE)*****
        ########################################################################
        #                           END OF YOUR CODE                           #
        ########################################################################

        return x
```

得到的模型训练、测试的损失曲线和测试的准确率曲线如下：

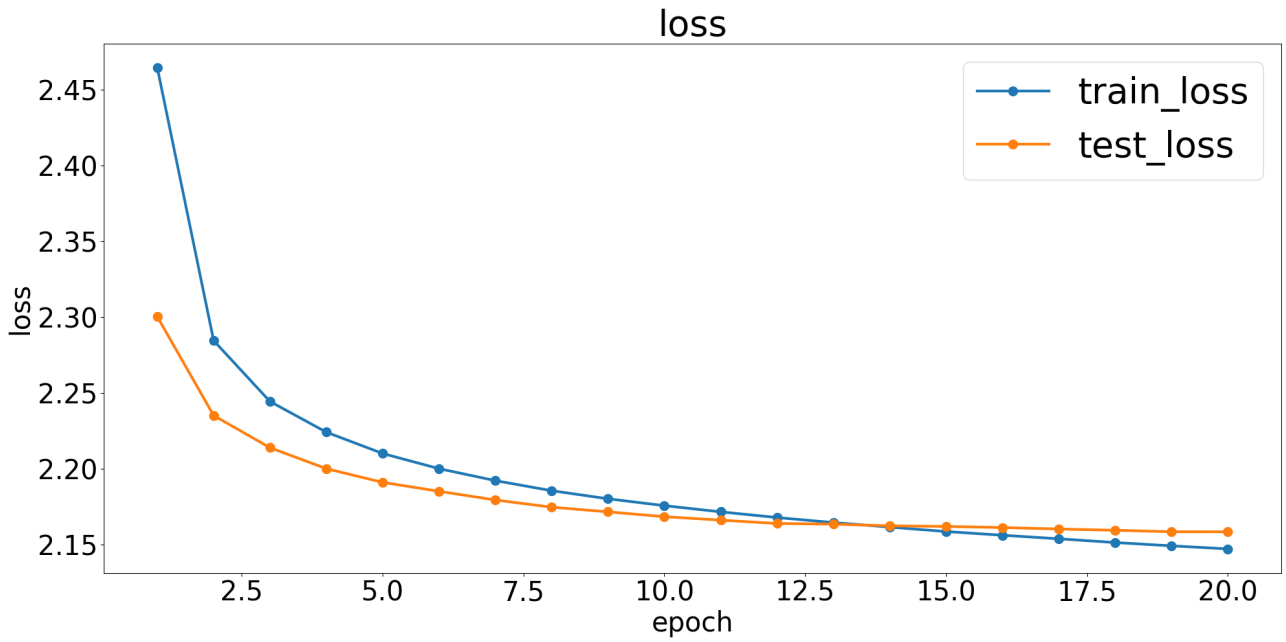


图 1: Loss 曲线

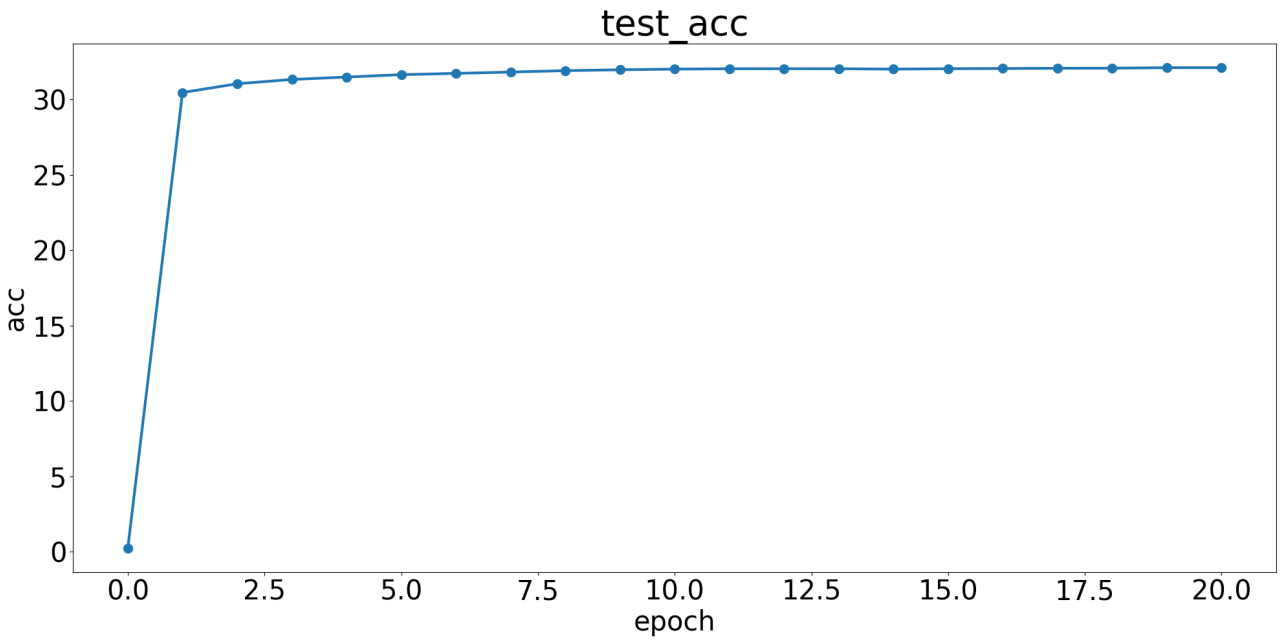


图 2: Acc 曲线

1.2.2 TASK 4

请完成 model.py 中的 PositionalEncoding 类并将其加入到 HarryPotterTransformer 类中。(完成代码即可，不用在报告中写文字说明) 请基于构建的网络完成训练，绘制训练、测试的损失曲线和测试的准确率曲线，并且分析 positional encoding 的作用。(请绘制在报告中)

代码如下:

```
class PositionalEncoding(nn.Module):
    def __init__(self, d_model, max_len = 5000):
        super().__init__()
        pe = None
        ################################################################################
        # TODO: compute the positional encoding                                        #
        ################################################################################
        # *****START OF YOUR CODE (DO NOT DELETE/MODIFY THIS LINE)*****

        pe = torch.zeros(max_len, d_model)
        position = torch.arange(0, max_len).unsqueeze(1)
        div_term = torch.exp(torch.arange(0, d_model, 2) * (-math.log(10000.0) / d_model))
        pe[:, 0::2] = torch.sin(position * div_term)
        pe[:, 1::2] = torch.cos(position * div_term)

        # *****END OF YOUR CODE (DO NOT DELETE/MODIFY THIS LINE)*****
        ################################################################################
        #                            END OF YOUR CODE                                  #
        ################################################################################
        self.register_buffer('pe', pe)

    def forward(self, x):
        """
        Arguments:
```

```
            x: Tensor, shape [batch_size, seq_len, embedding_dim]
        """
        x = x + self.pe.unsqueeze(0)[:, :x.size(1)]
        return x

class HarryPotterTransformer(nn.Module):
    def forward(self, x):
        attn_mask = None # you can omit this for Task 4 and Task 5
        ##########################################################################################
        # TODO: finish the forward pass                                                          #
        ##########################################################################################
        # *****START OF YOUR CODE (DO NOT DELETE/MODIFY THIS LINE)*****

        x = self.embedding(x) # [batch_size, seq_len, feature_size]
        x = self.pos_encoding(x) # [batch_size, seq_len, feature_size], Added in Task 4
        x = x.permute(1, 0, 2) # [seq_len, batch_size, feature_size]
        x = self.transformer_encoder(x, mask=attn_mask) # [seq_len, batch_size, feature_size]
        x = x.permute(1, 0, 2) # [batch_size, seq_len, feature_size]
        x = self.decoder(x) # [batch_size, seq_len, vocab_size]

        # *****END OF YOUR CODE (DO NOT DELETE/MODIFY THIS LINE)*****
        ##########################################################################################
        #                             END OF YOUR CODE                                           #
        ##########################################################################################

        return x
```

得到的模型训练、测试的损失曲线和测试的准确率曲线如下：

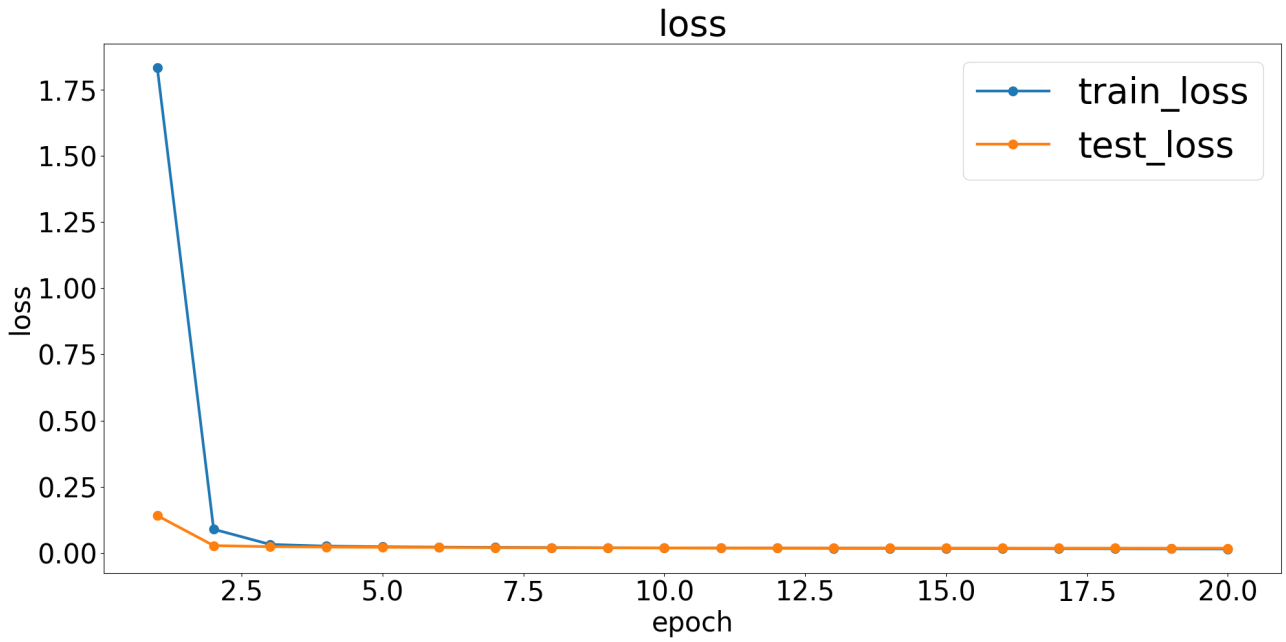


图 3: Loss 曲线

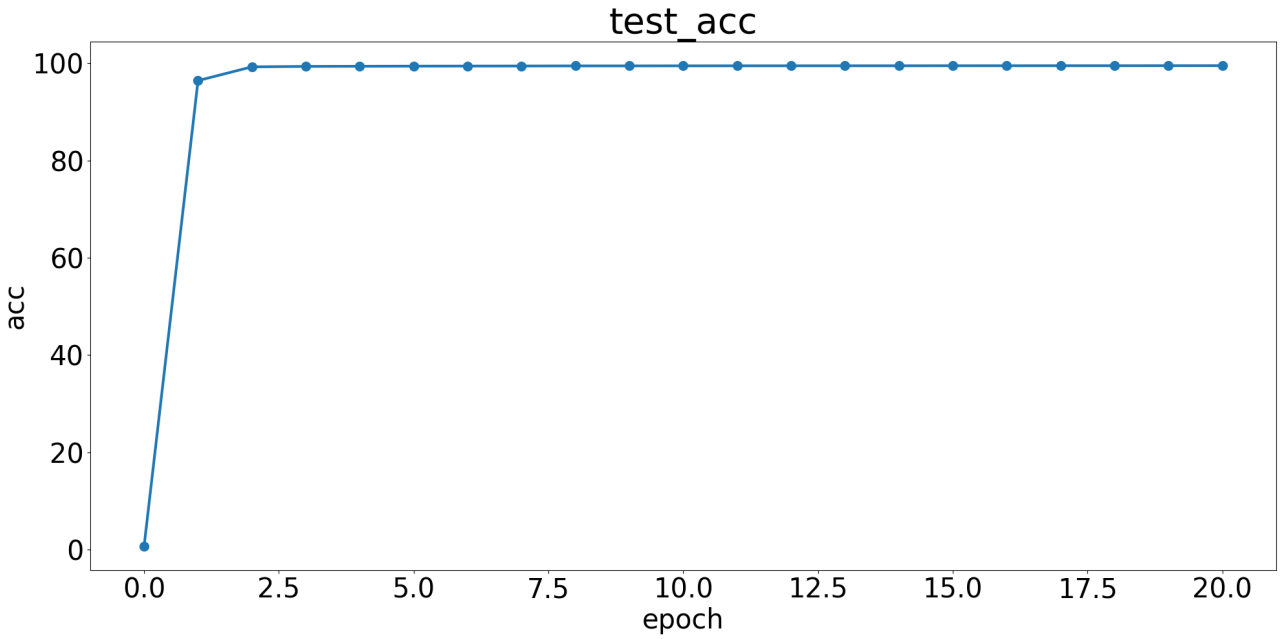


图 4: Acc 曲线

可以看到再使用了 PositionalEncoding 后，Loss 曲线下降显著加快，PositionalEncoding 为输入序列中的每个位置引入位置信息，从而弥补自注意力机制对序列顺序缺失的不足。但是曲线还是很奇怪，Loss 下降速度过快，原因是因为没有使用 attention mask 防止训练和测试时使用未来的文本数据。

1.2.3 TASK 5

请实现 attention mask 并将其加入到 HarryPotterTransformer 类的 forward 函数中。（完成代码即可，不用在报告中写文字说明）请基于构建的网络完成训练，绘制训练、测试的损失曲线和测试的准确率曲线，并且分析 attention mask 的作用。（请绘制在报告中）

代码如下：

```
class HarryPotterTransformer(nn.Module):
    def forward(self, x):
        attn_mask = None # you can omit this for Task 4 and Task 5
        ################################################################################
        # TODO: finish the forward pass                                                #
        ################################################################################
        # *****START OF YOUR CODE (DO NOT DELETE/MODIFY THIS LINE)*****

        x = self.embedding(x) # [batch_size, seq_len, feature_size]
        x = self.pos_encoding(x) # [batch_size, seq_len, feature_size], Added in Task 4
        x = x.permute(1, 0, 2) # [seq_len, batch_size, feature_size]
        attn_mask = torch.triu(torch.ones(x.size(0), x.size(0)), diagonal=1).to(x.device) # Added in Task 5
        attn_mask = attn_mask.masked_fill(attn_mask == 1, float('-inf')) # Added in Task 5
        x = self.transformer_encoder(x, mask=attn_mask) # [seq_len, batch_size, feature_size]
        x = x.permute(1, 0, 2) # [batch_size, seq_len, feature_size]
        x = self.decoder(x) # [batch_size, seq_len, vocab_size]

        # *****END OF YOUR CODE (DO NOT DELETE/MODIFY THIS LINE)*****
        ################################################################################
```

```
        #                           END OF YOUR CODE                            #
        ###############################################################################

        return x
```

得到的模型训练、测试的损失曲线和测试的准确率曲线如下：

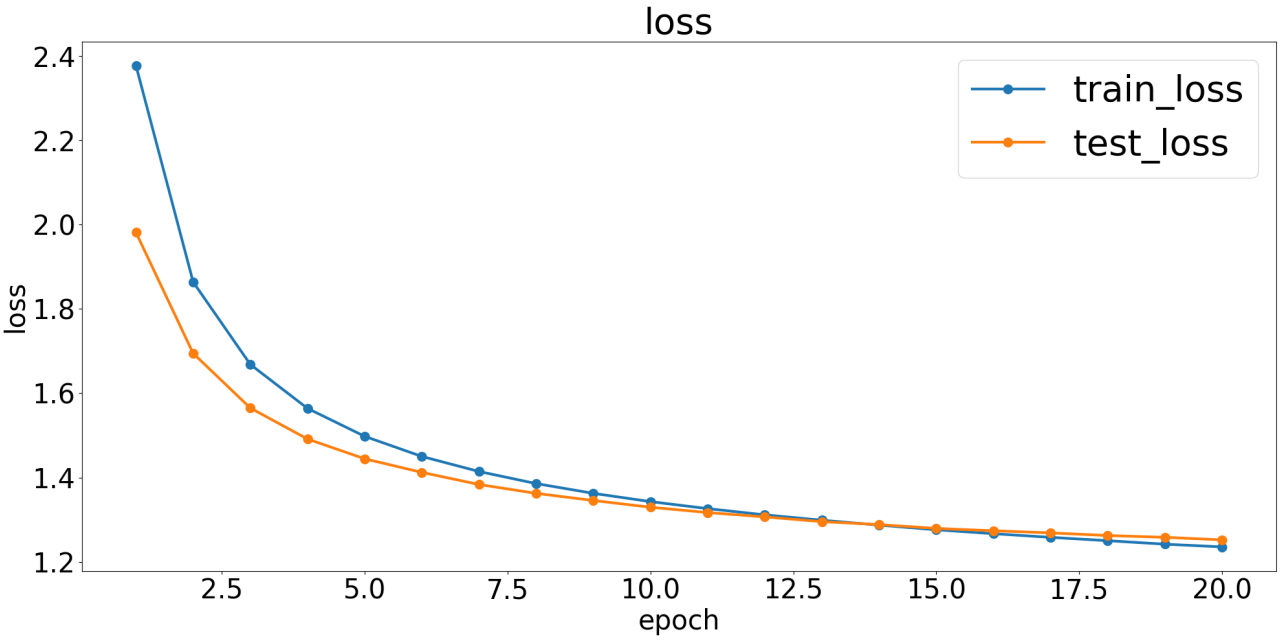


图 5: Loss 曲线

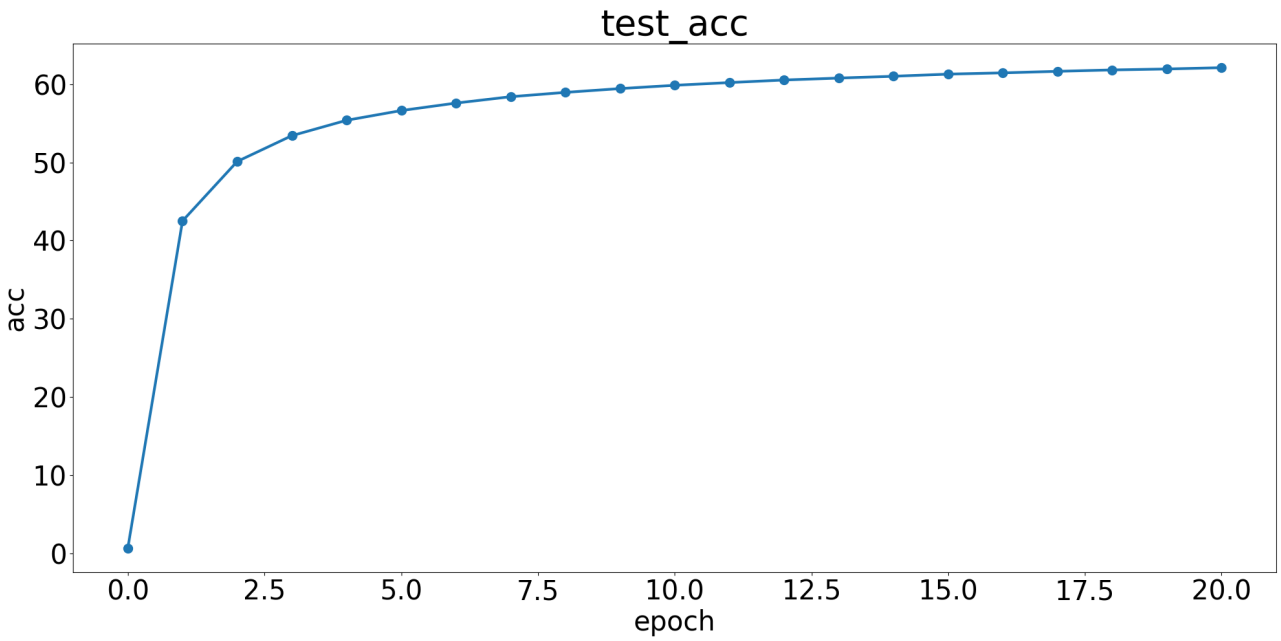


图 6: Acc 曲线

使用 attention mask 之后，Loss 曲线变得更加正常，没有急速下降。attention mask 的作用是防止模型访问未来时间步的信息。如果模型访问了未来时间步的信息，将导致目标值泄露，破坏训练。

1.3 文本生成

1.3.1 TASK 6

在默认参数（temperature=1, strategy='sampling'）下，调整模型生成使用的 seed_words，将你最喜欢的生成结果附在报告中。

seed_words 调整如下：

```
# generate
# Try different seed_words. Find interesting ones.
seed_words = "Harry␣Potter␣came␣to␣Tsinghua␣University␣and␣attended␣a␣course␣named␣Introduction␣to␣
    Auditory␣Visual␣Information␣System.␣"
# Experiment with different temperatures to observe their impact on diversity and stability.
```

得到的生成如下：

图 7: 输出

1.3.2 TASK 7

分别调整模型生成使用的 temperature、strategy，观察输出段落，你是否发现一些规律？请简单描述不同温度和不同策略下的结果差异，并简单分析原因。

分别调整 temperature、strategy，得到的一部分输出结果如下：

- temperature=0.8, strategy=sampling

检测到英语 - 中文 术语库 通用场景

Harry Potter came to Tsinghua University and attended a course named Introduction to Auditory Visual Information System. Yes, then, I can't stand they were mouth of once for your father and that will entered through the table inluder his mother books with wave of silk in the side of his fist, and he stared out the wall trick and spite of the rain he grasped happily to for the secrets, he had always where the desk and dangerous great single moment of him, and he had told Harry continued when he saw the table to brough the way back at once, and she pulled out his stupid to his face and hands, simply around the floor, pulling his own from his partment and threat the stands, his eyes on her wand unseated to be with the marbles minutes and now that strolled the classroom ran the short of doing the potion of silvery shocked by the room was laughing a face of which mates with everyone else's clear and then fall that between and pug his always clutching an' the get would. That this is searchest or enough to Ginny and for some that Malfoy were business going to shriek of his telescope. "Listen, Harry and Ron stepped that he would not take it; Harry had not seen outside their indignant of the room, he saw Harry and Ron placed out of his fist and door behind her books, looking at Harry. . . . ." "It said it quietly," said "She appeared, sir, "I don't know." Harry strided alarmed at Hogwarts were he had some as a book in the country of Magic had on his jaggered again and a potion of great-green and setting a burst of her early dirty to know what he was right being him that he was going to be something you out of restrated to the floor, it was almost the old hand turned around in the air. The floor; then killed in the raguin. "You heard you to be are of the Hall girl soul anger that made sure you just want your test of time? Oh dear," said George starily. "Just can be unfortunately, he's not in the rest of the Potions idea of deeply, a room. Horace he his wand up called the cloak crashed and set the three laying the room, pumpkin about the time times and more generating than to go to rest of th

重点词汇 AI润色 2120/5000

哈利·波特来到清华大学，参加了一门名为《听觉视觉信息系统概论》的课程。是的,然后,我受不了他们的嘴一旦为你父亲和进入通过与波表inluder母亲书丝绸的拳头,他盯着墙上的技巧,尽管下雨他抓住幸福的秘密,他一直在他的书桌和危险的伟大的一刻,他告诉哈利继续当他看到桌上布拉夫的方式回到曾经,她掏出他的愚蠢的他的脸和手,他只是在地板上转了一圈，从他的房间里抽出自己的魔杖，威胁着看台，他的眼睛盯着她的魔杖，把它从椅子上拿下来，和弹子弹珠在一起。现在，他在教室里溜达了几分钟，差点就喝了银色的药水，被房间里的人吓得大笑起来，他的脸和其他人的脸都很清楚，然后落在中间，巴哥总是紧紧抓住他的手。对金妮来说，这已经足够了，对一些人来说，马尔福的望远镜会让他尖叫起来。“听着，哈利和罗恩说他不会接受的。哈利没有看到他们愤愤不平的房间外，他看见哈利和罗恩从他的拳头和门后面被放在她的书后面，看着哈利. . . . . .”“它悄悄地说,”她出现了，先生，“我不知道。”哈利在霍格沃茨惊恐地大步走着——他身上又带了一本魔法国的书，又带了一瓶深绿色的药水，又带了一阵她早期的污秽，知道他是对的——他要变成你想要的东西——他倒在地上，几乎是那只苍老的手在空中转了个身。地板上;然后在废墟中被杀。“你听说你是霍尔女孩的灵魂愤怒，所以你才想要接受时间的考验？”哦，天哪。”乔治不耐烦地说。只是可不幸的是，他并没有在魔药课的其他方面有很深的想法，一个房间。霍勒斯举起魔杖，叫斗篷啪的一声摔了起来，让三个人躺在房间里，南瓜约了时间几次，更生不如去休息了

以上翻译结果来自有道神经网络翻译（YNMT）· 通用场景

检测到英语 - 中文 术语库 通用场景

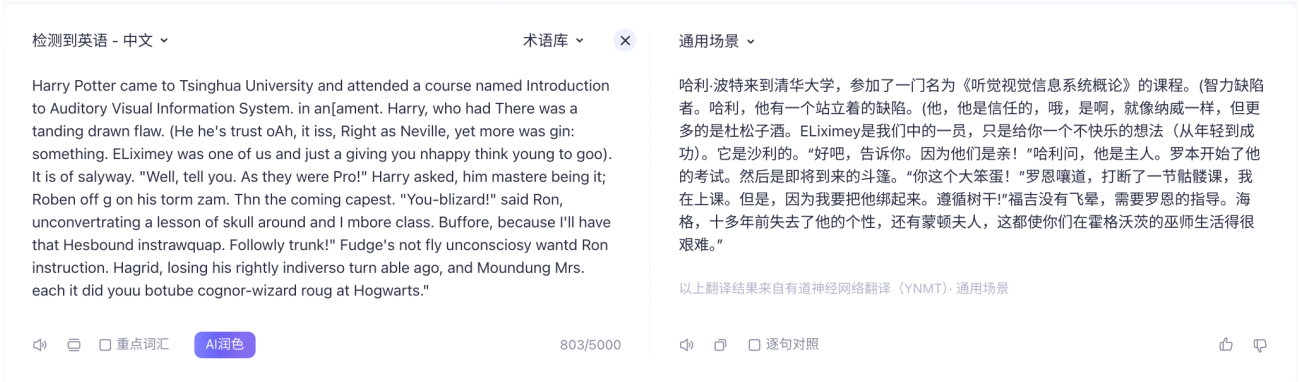
Harry Potter came to Tsinghua University and attended a course named Introduction to Auditory Visual Information System. These stood of homework, his ears widened the Ministry of Magic with-ting class and given the security that he was staying to be annooying seat on out of its house-elf can with Harry spiders with his jean to flush out first years. You will us were hand it too. He sneered Ginny, and he was falling as though he was still in the further! I can decorations what was a little stories and were so still for a matter with your had come to see Slughorn to his safety. Harry are to his arm was quite number out of the side of the Prince left that Harry was corner he had already to help to Dumbledore load of the corry, the piece of parchment, but he searched his wand in out of his death one of the particular memory had not to take a seat expression that you were in the buors of your age rest to have dressed out that he had been able to the staircase in the other, stomach on the wringe, Harry saw all of their full. Harry and Hermione not wanted to go from the basin, and he began though for a house-elf people me. It was a grave of his wand; then were in the middle in antidote departed, even handing his wand in the room was new to circle that the Dursleys, his mind that he made a rubbied it: the truth?" he added, pulling his face between the "art of the Pumpkin is searched stuff. "I don't know what you inter that you now?" "Yes," said Harry in a sharp. "You have this and must be over the second of of the subjects, his wand; he knew was wearing that what Harry had the grounds of light, and it was now about that raised out his arm, his low of grind pudding glass and the hospital wing him into the days' eyes had been for him. "You can need to pay the portrait was almost friends of the staff when he wanted to start you at the would all the third that the other whole legs out of the pools had been completely hit and he couldn't mind have me that she was tightly over his eyes spells of the corner. "The Mundungus of them unexpected the other class of the place was a flaming more chair. I

重点词汇 AI润色 2119/5000

哈利·波特来到清华大学，参加了一门名为《听觉视觉信息系统概论》的课程。在一大堆家庭作业之后，他的耳朵在魔法部的课堂上竖起来，考虑到他留下来的安全问题，他要和哈利一起坐在它的家养小精灵罐子里，让人讨厌，蜘蛛用他的牛仔裤把一年级新生赶出去。你也会把它交给我们的。他对金妮冷笑了一声，然后往下掉，好像他还在远处！我可以把刚才的一些小故事说出来，你刚才那么安静，是来给斯拉格霍恩送行的。哈利是他的手臂相当数量的的哈里王子离开了那个角落里他已经帮助邓布利多负载科里,一张羊皮纸,但他搜查了他的魔杖从他的死亡的一个特定的内存没有坐下buors表达你的你的年龄穿出,他已经能够楼梯,胃wringe,哈利看到他们所有的全部。哈利和赫敏都不愿意离开脸盆，于是他开始替一个家养小精灵说话。这是他魔杖的坟墓；然后就在解药的中间离开了，连把魔杖递到房间里都是新圈的那个德思礼家，他心里想他做了一个揉了揉它：“真相吗？”他补充道，把脸拉在“艺术南瓜是被搜过的东西”之间。“我不知道你现在在说什么？”“是的。”哈利尖声说。“你拿着这个，必须越过第二个科目——他的魔杖；他知道他戴的是哈利身上的那件东西，那是光明的基础，现在他抬起了手臂，他的磨碎的布丁镜和医院的翅膀进入了白天的眼睛。“你可以需要付那幅肖像几乎是教工的朋友，当他想要开始你的时候，他发现其他的整条腿都从水池里完全被击中了，他不介意我让她紧紧地遮住他的眼睛。他们中的蒙顿格斯没有想到，这里的另一个班是一把燃烧着的椅子。我

以上翻译结果来自有道神经网络翻译（YNMT）· 通用场景

- temperature=1.4, strategy=sampling



- temperature=0.8, strategy=greedy

- temperature=1.4, strategy=greedy

通过比较可以发现，调整 temperature 和 strategy 可以改变输出内容的随机性。当 temperature 较低时，输出内容更加稳定，模型更倾向于选择概率最高的词，输出较为保守，创意性不足，但文本通常更流畅，具体体现在生成的段落更加通顺；当 temperature 较高时，输出内容更加随机，可能生成一些意外的、有创意的内容，同时可能会出现不连贯或语法错误，具体体现在生成的段落语法出现更多问题；strategy 的选择会影响输出的随机性，sampling 策略会根据 temperature 的大小随机选择下一个词，greedy 策略会选择概率最高的词，生成内容通常连贯但缺乏多样性，容易陷入重复或固定模式。