CS103_Fall Project

主题：AI诗人

小组成员：虞快 黄增荣 杨家鉴 徐璟源

1. 研究背景

当前，人工智能被利用到各个领域，且表现出了十分优秀的效果，但在某些方面，例如中国文化中的诗词部分，诸如ChatGPT等的AI在诗歌体裁、韵律以及主题等各个方面的表现均不能满足要求。因此，针对这些部分进行AI设计是十分有必要的。当前，国内已经出现了一些以满足中国诗词创作为背景的模型。

2. 模型复现

ChatGPT

欢度元旦·其一
——GPT4
欢度元旦，喜气盈门新岁到。
笑声洒满街巷间，人海翻腾如潮。
旧忧一扫尽，希望如灯照。
携手迎新春，万象更新悄.

评价：不是古诗。

欢度元旦·其二
——GPT4
欢庆佳节迎新年，
雪舞风歌度元旦。
万家灯火共欢笑，
四季平安祈愿圆。

评价:

- 字数符合要求，押韵未符合要求。

Mengzi (based on Transfer Text-to-Text Transformer)

作诗:

- 欢度元旦

模仿:

- 李清照

诗歌:

年年爆竹报新晴，

喜气先随节物生。

莫道今宵无此景，

且看明日是新晴。

平仄检查:

年年爆竹报新晴

古: 平平仄仄仄平平

平平仄平仄平平

喜气先随节物生

古: 仄庆平平仄仄平

今: 仄仄平? 平仄平

莫道今宵无此景

古: 仄仄平平平仄仄

今: ? 仄平平平仄厂

目看明日是新晴

古: 仄通平仄人平平

今: 通通平仄仄平平

评价:

- 平仄正确，韵律符合，但会有用词重复，如“新晴”

模型分析:

- 该模型基于Mengzi预训练模型。

模型训练:

- 训练前将训练诗句数据进行预处理清洗，过滤部分不规范格式或错误字符
- 将剩余数据进行编码，统一数据集输入格式
- 传入transforms的T5ForConditionalGeneration模型进

清华九歌

平仄检查:

年来欢度不须猜，
古: 平平平仄通平平， ♦猜【十灰平声】
今: 平平平通仄平平，

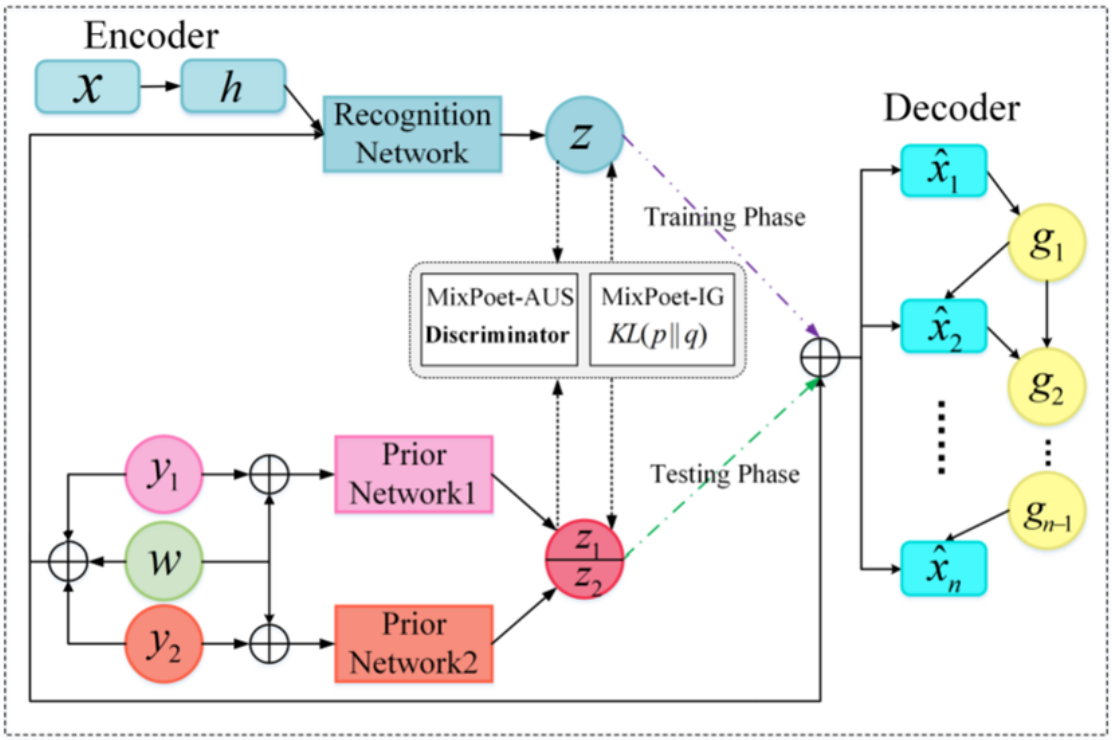
又见新阳换旧胎。
古: 仄仄平平仄仄平。 ♦胎【十灰平声】
今: 仄仄平平仄仄平。

自有一般堪慰藉，
古: 仄仄仄平? 仄仄， ♦藉【二十二祃去声】 ♦藉【十一陌入声】
今: 仄仄通平平仄通，

明朝春去却还开。
古: 平平平仄仄平平。 ♦开【十灰平声】
今: 平平平仄仄平平。

评价:

- 平仄对应，韵律正确，诗词中提到“新旧更替”“欢度”符合关键词。



训练办法：

- 半监督学习+对抗学习
- 评价因素：
  - 流畅性（是否符合诗词结构）
  - 上下文连贯性（主题与逻辑是否相符）
  - 意义（诗歌传达的信息）
  - 美感（是否有意境）
  - 主题相关性（是否符合给定主题）
  - 整体质量（对于是个的总体印象）

3. 环境配置

清华九歌：

- python>=3.7.0
- pytorch>=1.3.1
- sklearn>=0.19.2
- matplotlib>=2.2.3

4. 数据集预处理

清华九歌：

- 将THU-CCPC数据集中的训练集、验证集和测试集添加到*MixPoet/preprocess/*目录中。
- 将THU-CRRD数据集中的pingsheng.txt、zesheng.txt、pingshui.txt和pingshui_amb.pkl文件添加到*MixPoet/data/*目录中。

Mengzi：

- Mengzi-T5 model （中文）
- ```
  from transformers import T5Tokenizer, T5ForConditionalGeneration
  tokenizer = T5Tokenizer.from_pretrained("Langboat/mengzi-t5-base")
  model = T5ForConditionalGeneration.from_pretrained("Langboat/mengzi-t5-base")
  ```

5. 执行命令

清华九歌：

- 在MixPoet/preprocess/目录下，只需运行以下命令：

  ```
  python preprocess.py --n 150000
  ```

  参数*n*表示用于半监督训练的未标记实例的数量。当使用提供的CQCF样本集时，我们建议*n*设置为大约150,000。对于更大的标记数据集，可以设置更大的*n*。

  运行preprocess.py之后，请将生成的vocab.pickle、ivocab.pickle、semi_train.pickle和semi_valid.pickle移动到MixPoet/corpus/目录中，并将test_inps.txt和training_lines.txt移动到MixPoet/data/目录中。

6. 训练

在 MixPoet/codes/ 中，运行：

```
python train.py
```

编码器和解码器将被预训练为去噪自动编码器，分类器将使用标记的诗歌进行预训练。然后，基于这些预训练模块对 MixPoet 模型进行训练。

也可以编辑 MixPoet/codes/**config.py** 来修改配置，例如隐藏大小、嵌入大小、数据路径、训练周期、学习率等。

在训练过程中，会输出一些训练信息，例如：

```
factor1 label: 1
factor2 label: -1, inferred: 1
key: 洞门
trg: 非凿非疏出洞门|源深流峄合还分|高成瀑布漱通客|清入御沟朝圣君
out_post: 非凿深深亦洞门|高深清水入难分|清风石布吟通客|别入玉衣朝圣君
out_prior: 一鹤曾窥一夜歌|两三应共对清光|不须惆怅高堂醉|却恨飞花洞里长

epoch: 19, 401/937 42.8%, 0.243 s per iter, lr: 0.0003, tr: 0.85, tau: 0.010, noise: 0.000
  ppl:27.5, rec loss: 3.315, entropy loss: -0.587, cl loss w: 1.706, cl loss xw: 0.279
  dis loss: 0.272, adv loss: 1.393, latent dist: 602.831, factors dist: 29.269
```

训练和验证信息保存在 MixPoet/log/ 中。

7. 具体生成

要在交互式界面中生成一首诗，请在 MixPoet/codes/ 中运行：
```

```
python generate.py -v 1
```

然后可以输入关键字、长度和因子标签，然后得到生成的 pome：

```
input a keyword:>烽火
specify the length, 5 or 7:>7
specify the living experience label
           0: military career, 1: countryside life, 2: other:, -1: not specified>-1
specify the historical background label
           0: prosperous times, 1: troubled times, -1: not specified>-1
inferred label1: 0
inferred label2: 0

generating step: 0

generating step: 1

generating step: 2

generating step: 3
烽火连天海气昏
胡笳吹尽塞尘喧
匈奴未灭关心壮
贾谊才堪报国恩
input a keyword:>
input a keyword:>月
specify the length, 5 or 7:>5
specify the living experience label
           0: military career, 1: countryside life, 2: other:, -1: not specified>0
specify the historical background label
           0: prosperous times, 1: troubled times, -1: not specified>0

generating step: 0

generating step: 1

generating step: 2

generating step: 3
城头雪初霁
楼角月徘徊
日暮征鞍去
河湟浊酒杯
input a keyword:>
```

通过运行：

```
python generate.py -v 1 -s 1
```

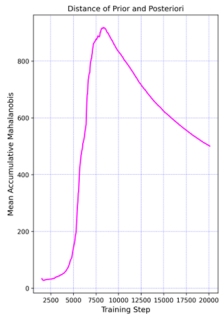
可以从候选光束中手动选择每条生成的线。

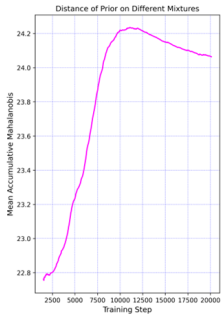
要使用包含一组关键字的输入测试文件生成诗歌，请运行：

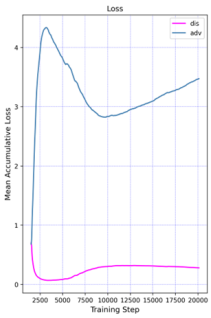
```
python generate.py -m file -l 5 -i ../data/test_inps.txt -o outs_5char.txt
```

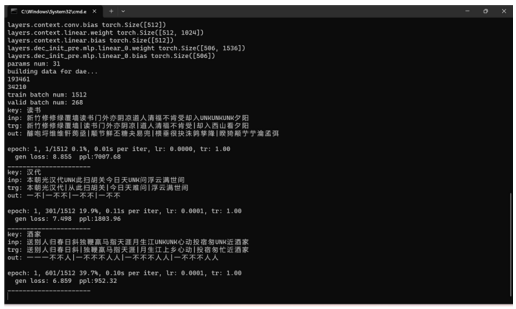
其中 l = 5 或 7，表示 5 条字符线或 7 条字符线的四行诗。

我们操作实验获得的：









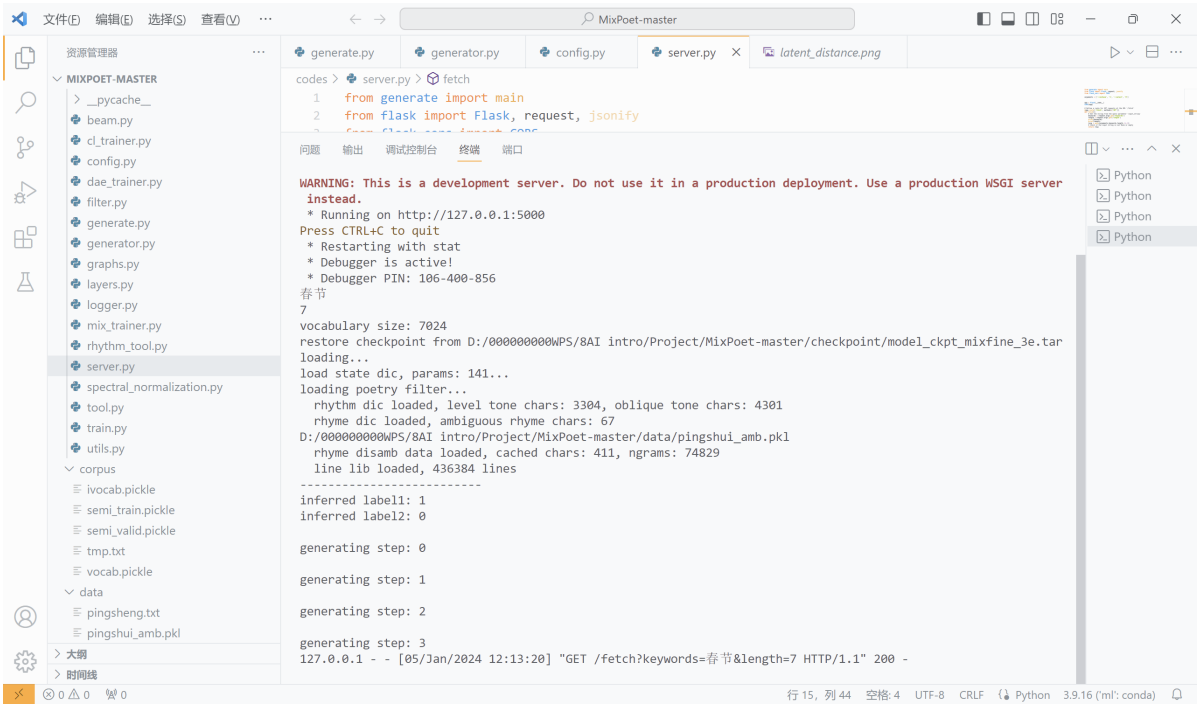
8.具体成品

前端页面：

登录展示界面

操作平台：

后端部分界面展示：



9. 展望

随着人工智能（AI）技术的迅猛发展，其在各个领域的应用不断拓展，为我们的生活带来了前所未有的便利和创新。在文学艺术领域，AI的介入也逐渐引起了广泛的关注与讨论。本研究以探索AI在文学创作中的独特应用为目标，致力于借助先进的计算机视觉技术和自然语言处理算法，构建了一个具有独创性的系统。该系统具备上传图片识别功能，并以此为基础生成富有诗意的配图与诗歌，结合对全唐诗、全宋词数据库的模糊匹配，为用户提供一种全新的文学创作体验。

通过本研究，我们成功地构建了一个融合先进技术与传统文学的创新系统。该系统在AI图像识别与创作生成领域取得了显著的成果，为用户提供了一种全新的文学创作体验。通过与全唐诗、全宋词数据库的有机结合，我们不仅在创作过程中注入了古典文学的精髓，也为现代文学创作带来了新的可能性。未来，我们将进一步完善系统性能，拓展适用范围，并深入探讨AI在文学艺术中的更广泛应用。本研究为文学与科技的融合开辟了一条新的道路，为未来的研究与应用提供了有益的参考。

10. 可能的解决方法

1. 优化深度学习算法：进一步优化系统中的深度学习算法，以提高图像识别的准确性和生成诗歌的质量。可以通过增加训练数据集、调整神经网络结构和优化超参数等手段，不断提升系统的性能。
2. 用户反馈与改进机制：引入用户反馈机制，建立用户体验的反馈循环。通过收集用户的评价和建议，及时调整系统的算法和功能，以更好地满足用户的需求，提高系统的实用性和用户满意度。
3. 多样性创作模式：扩展系统的创作模式，引入更多的文学体裁和风格，使用户能够体验到更加丰富多彩的文学创作。可以考虑加入散文、韵文等不同形式，以满足不同用户的创作偏好。优化深度学习算法：进一步优化系统中的深度学习算法，以提高图像识别的准确性和生成诗歌的质量。可以通过增加训练数据集、调整神经网络结构和优化超参数等手段，不断提升系统的性能。
4. 继续优化数据库匹配算法：深入研究改进全唐诗、全宋词数据库的模糊匹配算法，以提高系统对古典文学的敏感性和匹配准确性，使生成的诗歌更加融合传统文学的精华。
5. 优化深度学习算法：进一步优化系统中的深度学习算法，以提高图像识别的准确性和生成诗歌的质量。可以通过增加训练数据集、调整神经网络结构和优化超参数等手段，不断提升系统的性能。

11. 引用

[1] Z. Zhang, H. Zhang, K. Chen, Y. Guo, J. Hua, Y. Wang, and M. Zhou, “Mengzi: Towards lightweight yet ingenious pre-trained models for chinese,” 2021.

[2] Xiaoyuan Yi, Ruoyu Li, Cheng Yang, Wenhao Li and Maosong Sun. 2020. MixPoet: Diverse Poetry Generation via Learning Controllable Mixed Latent Space. In Proceedings of The Thirty-Fourth AAAI Conference on Artificial Intelligence, New York, USA.